(2)

AD_____

REPORT NO. T9-91

DATA ACQUISITION SYSTEM FOR COLLECTING BIOPHYSICAL AND PHYSIOLOGICAL DATA

U S ARMY RESEARCH INSTITUTE
OF
ENVIRONMENTAL MEDICINE
Natick, Massachusetts

UNITED STATES ARMY
MEDICAL RESEARCH & DEVELOPMENT COMMAND

The findings in this report are not to be construed as an official Department of the Army position, unless so designated by other authorized documents.

DISPOSITION INSTRUCTIONS

Destroy this report when no longer needed.

Do not return to the originator.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No 0704-0188

1. AGENCY USE ONLY (Leave blank)	2. REPORT DATE	3. REPORT TYPE AND DATES COVERED
	April 1991	Technical Report

4. TITLE AND SUBTITLE

Data Acquisition System for Collecting Biophysical and Physiological Data

5. FUNDING NUMBERS

62787A
3M162787A878 AE
DA335670

6. AUTHOR(S)

Stephen KW. Chang, MSEE., Ph.D.

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

U.S. Army Research Institute of Environmental Medicine
Natick, MA 01760-5007

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Same as 7.

10. SPONSORING/MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES

12a. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT (Maximum 200 words)

This report describes the Biophysics and Biomedical Modeling Division data acquisition system (AC128) and also serves as an operations manual. The AC128 is designed as a general purpose system capable of collecting 128 channels of data during biophysical and physiological protocol studies. The AC128 contains both hardware and software components. In this report, the HARDWARE section details the overall system design, component layout, and necessary hardware modifications. Hardware calibration and maintenance procedures are also included. The SOFTWARE section describes the system software programs. The software adheres to a modular design. The entire software code is included as Appendix A. The software programs were written using primarily QuickBASIC version 4.0. The size of the compiled executable file is approximately 150,000 bytes. The SOFTWARE section also serves as an operations manual. The manual guides the operator step-by-step through the system setup procedures and the data acquisition phase. Storage of acquired data (in ASCII format) for future processing is also described.

14. SUBJECT TERMS

General purpose data acquisition system; Operations manual; 128 channels; Biophysical and physiological data

15. NUMBER OF PAGES

242

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT	18. SECURITY CLASSIFICATION OF THIS PAGE	19. SECURITY CLASSIFICATION OF ABSTRACT	20. LIMITATION OF ABSTRACT
Unclassified	Unclassified	Unclassified	

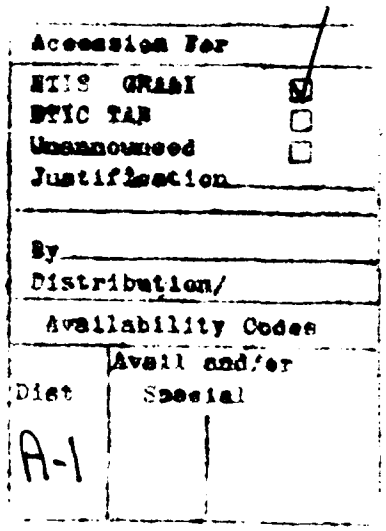
AD_____

Report No. _____

DATA ACQUISITION SYSTEM FOR COLLECTING BIOPHYSICAL AND PHYSIOLOGICAL DATA

by

Stephen KW. Chang, MSEE., Ph.D.



U. S. Army Research Institute of Environmental Medicine

Natick, MA. 01[illegible]0-5007

CONTENTS

	Page
EXECUTIVE SUMMARY	1
INTRODUCTION	3
HARDWARE	4
SETUP	6
Memory Requirement and Ramdisk	6
Analog-to-Digital Board	8
General Purpose Board	11
Thermistor Board	14
Inter-Connection Board	17
INPUT TYPES and MODULES DESCRIPTION	20
Voltage	20
Millivoltage	23
Wide Bandwidth Voltage and Millivoltage	23
Thermocouple	24
Current Input	25
RTD	25
Strain Gauge	25
Thermistor	26
HARDWARE MODIFICATIONS	28
Inter-Connection Board	28
Thermistor Board	28
CALIBRATIONS and MAINTENANCE	29
Analog-to-Digital Board	29
Thermistor Board	30
SOFTWARE	32
INITIAL SETUP	40
MAIN SELECTION	43
Load Setup Screen - F1	43
Save Channel Setup - F5	47
Save Collected Data - F7	47
DOS Shell - F8	55
Main Selection HELP - F10	55
Modify Current Active Setup - F4	55
New Setup File - F2	55
Acquire Data - F9	55

Quit AC128 Program - ESC 55
CHANNEL SETUP 58
Setup Completed 58
Device Name 61
Enter Device Name - F1 61
Remove Device Name - F8 61
Setup Completed - F9/ESC 61
Device Name HELP - F10 61
Channel Select 65
Enable Data Channel - F5 65
Disable Data Channel - F8 65
Setup Completed - F9/ESC 65
Channel Selection HELP - F10 65
Data Acquisition Timing Parameters - F1 68
Activate Data Plotting - F3 70
Deactivate Data Plotting - F4 70
Input Voltage and Gain Select 74
Input Voltage Range ±5V - F1 74
Input Voltage Range ±500mV - F2 74
Input Voltage Range ±50mV - F3 74
Input Voltage Range ±10mV - F4 74
Remove Input Voltage Selection - F8 74
Setup Completed - F9/ESC 74
Input Gain HELP screen - F10 74
Conversion Factor 78
Remove Conversion Factor - F8 78
Setup Completed - F9/ESC 78
Conversion Factor HELP - F10 78
Thermocouple - F1 81
Thermistor - F2 84
Linear Conversion - F3 89
Polynomial Conversion - F4 91
Unit Label 93
Enter Unit Label - F1 93
Remove Unit Label - F8 93
Setup Completed - F9/ESC 93
Unit Label HELP screen - F10 93
DATA ACQUISITION 97
Select/Change Printer - F4 97

Save Collected Data - F7 104
Start A New Study - F9 104
Exit Data Acquisition - ESC 104
Data Acquisition HELP - F10 104
Start Data Collection - F1/Space Bar 107
Place Event Marker - ^F10 114
Change Display Screen - Page Up/Page Down 114
Temporarily Suspend Data Acquisition - ESC/Enter 114
SYSTEM LIMITATIONS 116
DATA RESOLUTION 116
DATA PLOTTING CAPABILITY 117
APPENDIX -- AC128 Operating System Software Listing 118
Program Variable Declaration 119
Initial Setup module 123
Main Selection module 127
Data Acquisition module 147
Channel Setup module 166
Device Name submodule 169
Channel Select submodule 174
Timing 177
Data Plotting 180
Input Voltage and Gain Select submodule 186
Conversion Factor submodule 192
Thermocouple 194
Thermistor 195
Linear 197
Polynomial 198
Unit Label submodule 205
Utility subprograms 213
Assembly Language subroutines 232
DISTRIBUTION LIST 239

EXECUTIVE SUMMARY

This report describes the Biophysics and Biomedical Modeling Division general purpose data acquisition system (AC128) and also serves as an operations manual for the AC128. The AC128 is designed to collect up to 128 channels of data during biophysical and physiological protocol studies, using an IBM-AT compatible personal computer. The system is general purpose in that it can be tailored to suit the particular needs of individual protocols. The AC128 contains hardware and software components. Both the hardware setup and the software operations are described in this report.

The HARDWARE section describes the system hardware design, component layout, and individual hardware components of the AC128. Complete descriptions of the hardware modifications necessary for proper AC128 operations are also carefully detailed. Hardware calibration and maintenance procedures are included.

The SOFTWARE section describes the AC128 software programs. The software adheres to a modular design. In general, each main module is then subdivided into several submodules. Each submodule contains algorithms to handle a specific task. Detailed description of the programming algorithm in each module/submodule is included in this report. The entire software program code is included as Appendix A. The software programs were written primarily in BASIC language, using QuickBASIC version 4.0 editor/compiler. There are also small interface and data collection routines written in Assembly language. The size of the compiled executable file is approximately 150,000 bytes.

The SOFTWARE section also serves as an operations manual for the AC128 software system. The manual guides the operator step-by-step through the operating system. To meet the specific needs of a study, the system must be correctly configured. The manual describes how to set up the configuration. The manual also details the procedures of the data acquisition phase, and how the acquired data can be stored for future processing.

Keywords: General purpose data acquisition system, Operations manual, 128 channel inputs, Biophysical and physiological data

INTRODUCTION

The Biophysics and Biomedical Modeling Division general purpose data acquisition system (AC128) is designed to sample up to 128 channels of input data, using an IBM-AT compatible personal computer (PC). The most recent set of sampled data are displayed in real time on the PC monitor. All data are also temporarily stored in the PC memory. Options to save the collected data onto other more permanent storage devices are provided.

In this report the term "AC128" represents the entire data acquisition system, comprising both the hardware and the software. The software portion alone will be referred to as the SoS. The AC128 SoS does not replace the PC Disk Operating System (DOS). The SoS runs within the PC DOS and calls many DOS provided functions.

The primary types of input supported include thermocouples, thermistors, and voltages. All standard thermocouple types (T, E, J, K, B, R, S) are allowed. Thermistors which subscribe to the Steinhart-Hart conversion equation are supported. The ranges of voltage input can be from ±4.88 μV (μV=10^{-6}V) to ±10 V.

Nonstandard thermocouple and thermistor types which require special conversion algorithms can aiso be used. Secondary inputs including current input, RTD (resistance temperature detector), and strain gauges are also supported. Other input devices such as heart rate from a cardiac frequency/EKG monitor can be sampled provided the monitor supplies a voltage output proportional to the heart rate.

The primary design criterion of the AC128 software system (SoS) is a user-friendly interface. Channel configuration, data collection, and data storage are all integrated into a single system. The SoS allows the operator to configure each individual channel of the acquisition system. Completion of channel setup leads to data collection, and data storage. All the operations can be performed without leaving the AC128 SoS. HELP information is available at every stage to guide the operator through the operating procedures. Two research protocol studies have been conducted using the AC128. Experience gained from these studies indicates that the training period for operating the AC128 SoS is minimal. The program was written primarily using QuickBASIC, version 4.0, with small interface and data collection routines in Assembly language.

HARDWARE

Figure 1 displays the hardware system layout of the AC128. The hardware consists of components procured from two primary commercial sources (excluding the PC). The components are then modified to ensure compatibility and conformation to the AC128 system design. The AC128 is designed to accept a total of 128 channels of input data.

The first 96 channels (channels 0 - 95) are general purpose input channels. They consist of six general purpose input boards (Analog Devices 5B02), each is capable of accepting up to sixteen channels. The following types of input data are accepted: voltage, millivoltage, wide bandwidth voltage, wide bandwidth millivoltage, current, thermocouple, RTD, and strain gauge. The input ranges and characteristics will be discussed in detail shortly. Each channel can accommodate any one of the acceptable input types, thus allowing the investigators to mix and match according to their specific requirements. This degree of interchangeability is accomplished by using individual input modules to accept the specific input signals. Each module resides in one of the sixteen slots on the general purpose input board.

The last 32 channels (channels 96 - 127) are designed specifically to accept thermistor signals. A total of four thermistor boards (MetraByte EXP-RES) are used, each is capable of accepting up to eight thermistor inputs. Unlike the previous 96 channels, these 32 channels do not require any input module. The necessary circuitry is incorporated entirely on the thermistor board. These channels are dedicated to thermistor inputs and are functionally incompatible with the general purpose channels. Thermistors must be connected to these last 32 channels and not to the general purpose channels. The number of thermistor channels may be increased beyond thirty-two by adding additional thermistor boards. This will reduce the number of available general purpose channels by necessarily removing one or more of the general purpose boards. The total number of channels, general purpose plus thermistor, must remain less than or equal to 128 channels.

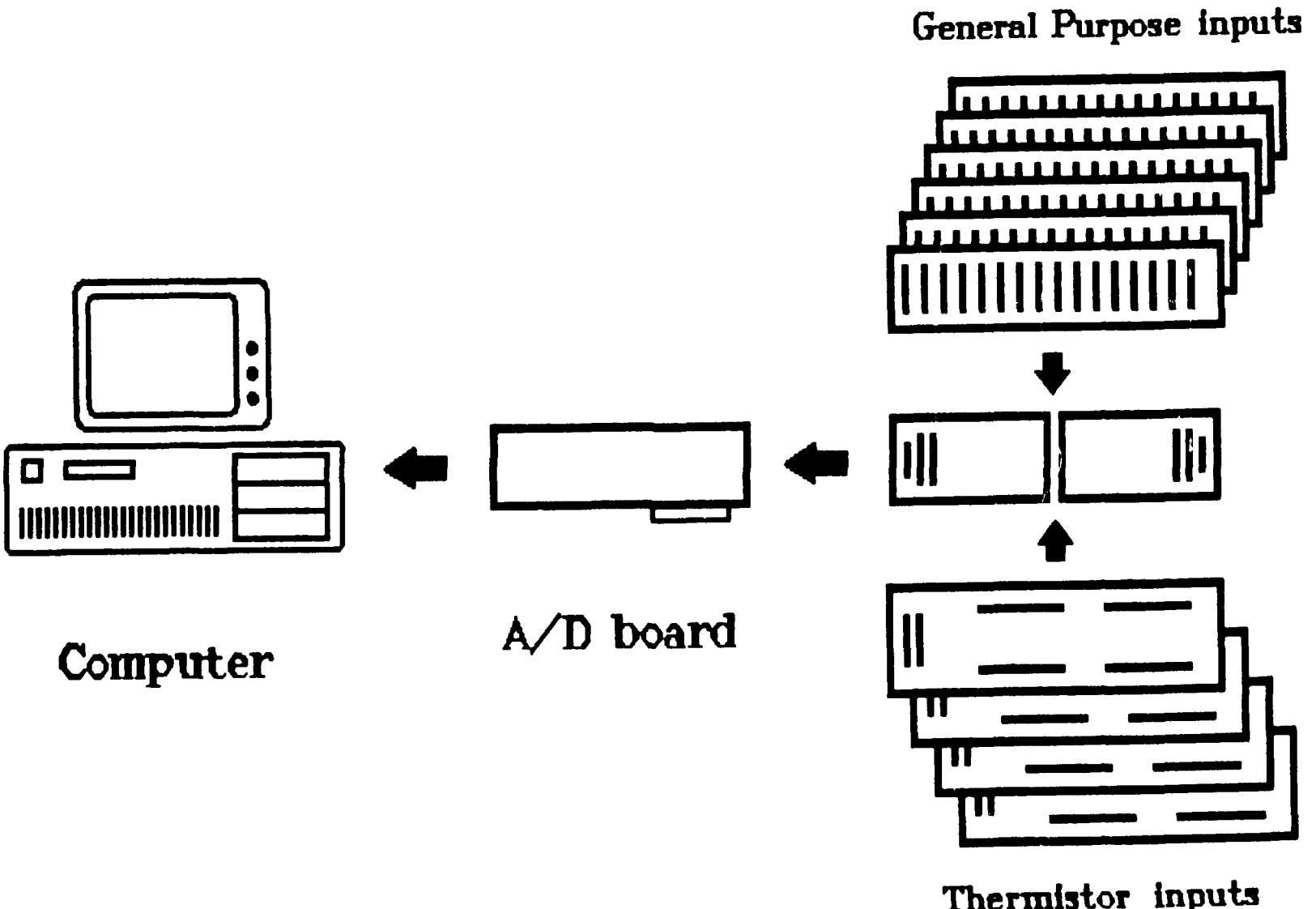


Figure 1. ACL2s hardware system layout

SETUP

Figure 1 is a schematic diagram of the AC128 hardware setup. Each hardware component will be discussed in detail. The path of information is as follows: analog data from the data collection boards (general purpose or thermistor boards) are brought together on the inter-connection boards, then routed to the analog-to-digital board (a PC plug-in board), and finally to the PC to be processed and displayed on the monitor screen.

The AC128 input boards are housed in a 19-inch mounting rack (see Figure 2). The maximum four thermistor boards are shown in Figure 2, whereas only four of the six maximum general purpose boards are included. The rack housing is mounted on casters for easy mobility.

Memory Requirement and Ramdisk

The AC128 SoS should be run on PC system equipped with 640 kB (kB=1000 bytes) of system random access memory (RAM). In addition, at least another 2MB (MB=1,000,000 bytes) of RAM is necessary to be configured as a ramdisk. The ramdisk may be configured using either PC extended or expanded memory. The SoS, by default, assumes the ramdisk is configured as logical device d:, although any other logical designation is acceptable. The ramdisk is used as a temporarily memory to hold the collected input data during data acquisition. Hard disk and floppy disk are too slow to serve this purpose.

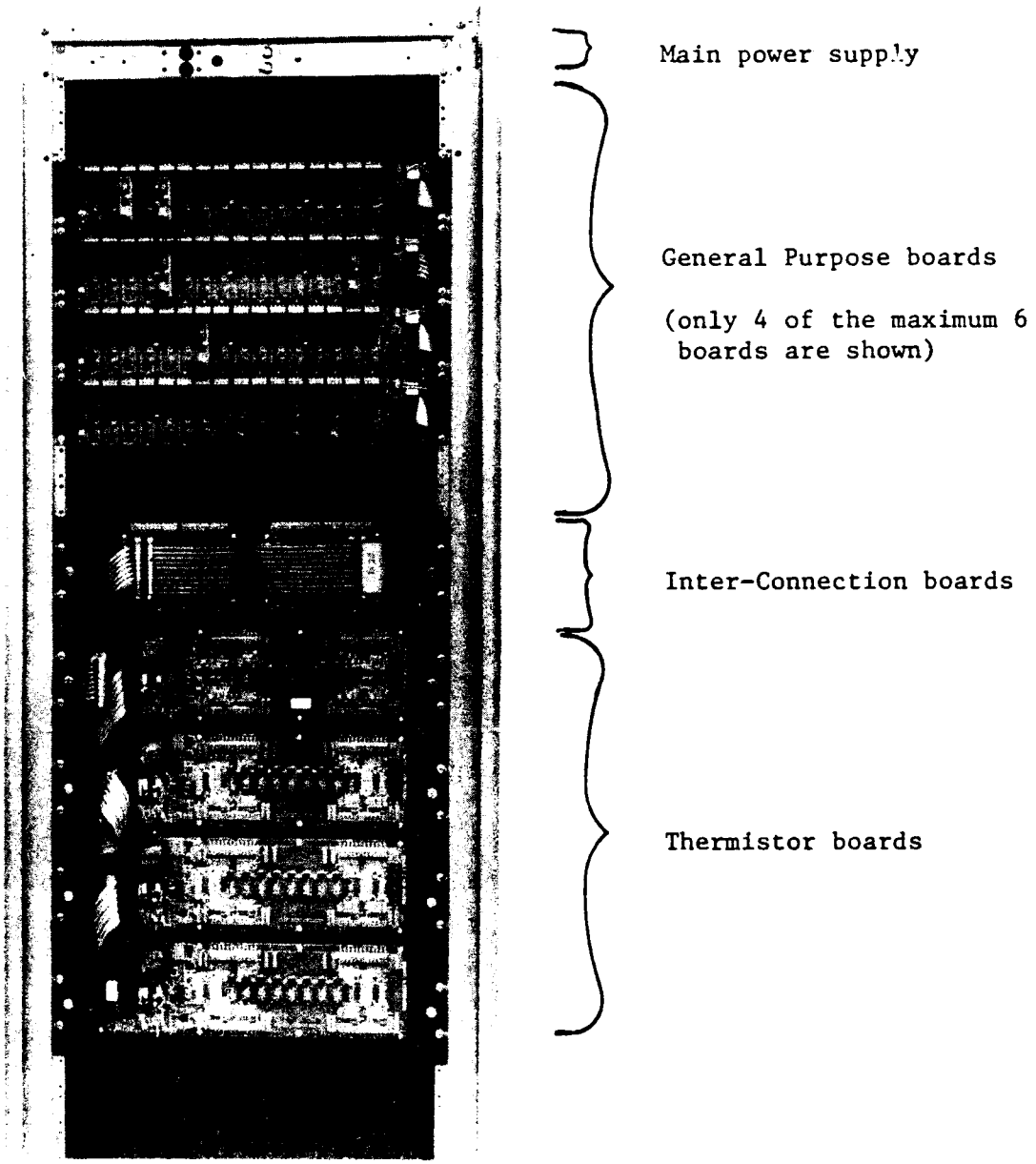


Figure 2 AC128 hardware system housing

Analog-to-Digital Board

The analog-to-digital (A/D) board (Analog Devices RTI-815-F, see Figure 3) is a PC plug-in board, requiring an 8-bit PC bus slot. The main function of the A/D board is to convert analog input signal to digital data. The digital data can then be accepted by the PC and processed by the AC128 SoS.

The internal registers on the A/D board are mapped to 16 bytes of the PC I/O memory address. The address of the starting byte is selected by switch S1 on the A/D board. The location of switch S1 is indicated on the A/D board layout schematic diagram of Figure 4. The AC128 uses the starting byte address of 330H (hexadecimal). The switch setting on S1 is shown in the following table. Notice, switch number 3 is not used.

switch S1 position	1	2	4	5	6
on/off setting	off	on	on	off	off

The A/D board itself has sixteen possible input addresses, from 0 to 15. Each data input board (general purpose and thermistor boards) must be mapped to one of these addresses. The general purpose boards are mapped to addresses 14 and 15. The thermistor boards are mapped to addresses 1 through 4. The address mapping will be explained in detail in the appropriate sections.

More detailed information about the A/D board operation can be found in the Analog Device RTI-815 User's Manual.

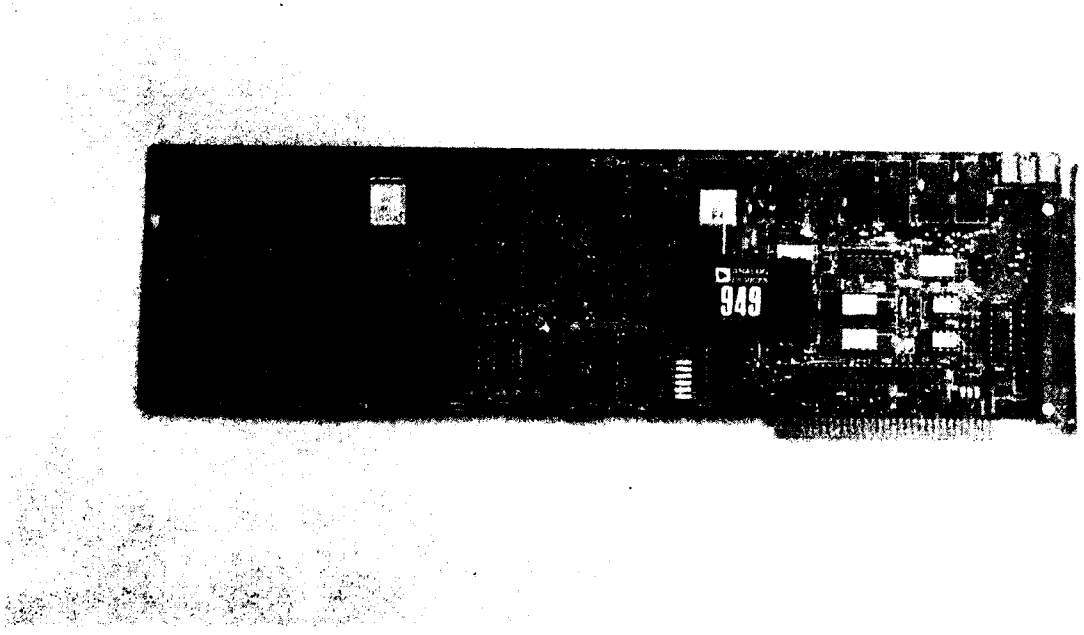


Figure 3 Analog-to-digital (AD) board

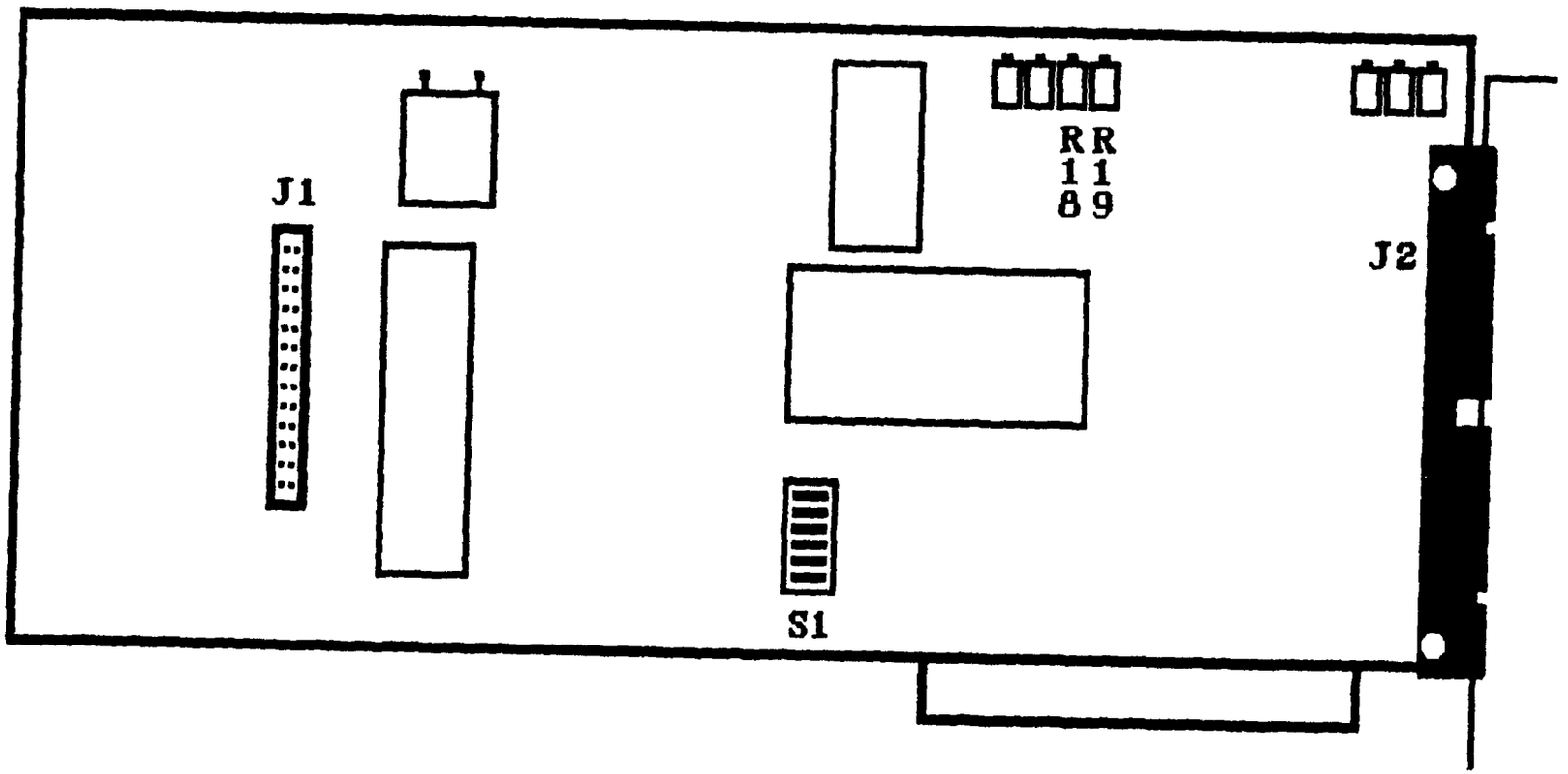


Figure 4 A/D board layout

General Purpose Board

The general purpose (GP) board (Analog Devices 5B02) is shown in Figure 5. Figure 6 gives a schematic of the GP board layout. There are sixteen slots shown, each capable of accepting one of the input modules described in the Input Module Description section below. On the GP board, each of the sixteen slots can be addressed by using one of the 64 (0–63) available addresses (the GP board address should not be confused with the 16 A/D board addresses discussed earlier). The SH1 and SH6 jumper blocks (see Figure 6) are used to select the GP board address range, as per the following table.

GP address range	SH1	SH6
0–15	5	10
16–31	4	9
32–47	3	8
48–63	2	7

The 64 available addresses allow four GP boards to be cascaded together. However, with a total of six GP boards in AC128, two cascade series are necessary. One GP cascade consists of four boards, and the other has two boards. The 4-board GP cascade uses all of the available 64 addresses, hence each GP board in this cascade series must use one of the jumper combinations in the above table. Duplications of jumper settings will cause severe input data conflicts. The 2-board cascade uses the 5-10 and 4-9 (GP address 0-31) jumper combinations. The GP boards are cascaded together using 3-connector, 26-wire, flat ribbon cables (Analog Devices CAB-01)

The two GP cascades are mapped to addresses 14 and 15 on the A/D board. The 4-board GP cascade is mapped to address 15, and the 2-board cascade is mapped to address 14. More information on the address mapping is provided in the discussion of the inter-connection board and the hardware modification sections.

Each GP board is equipped with its own isolated +5V power supply. Terminal block (TB) 16 (see Figure 6) accepts the +5V and ground (Gnd) supplies.

More detailed information about the GP board can be found in the Analog Device 5B Series User's Manual.

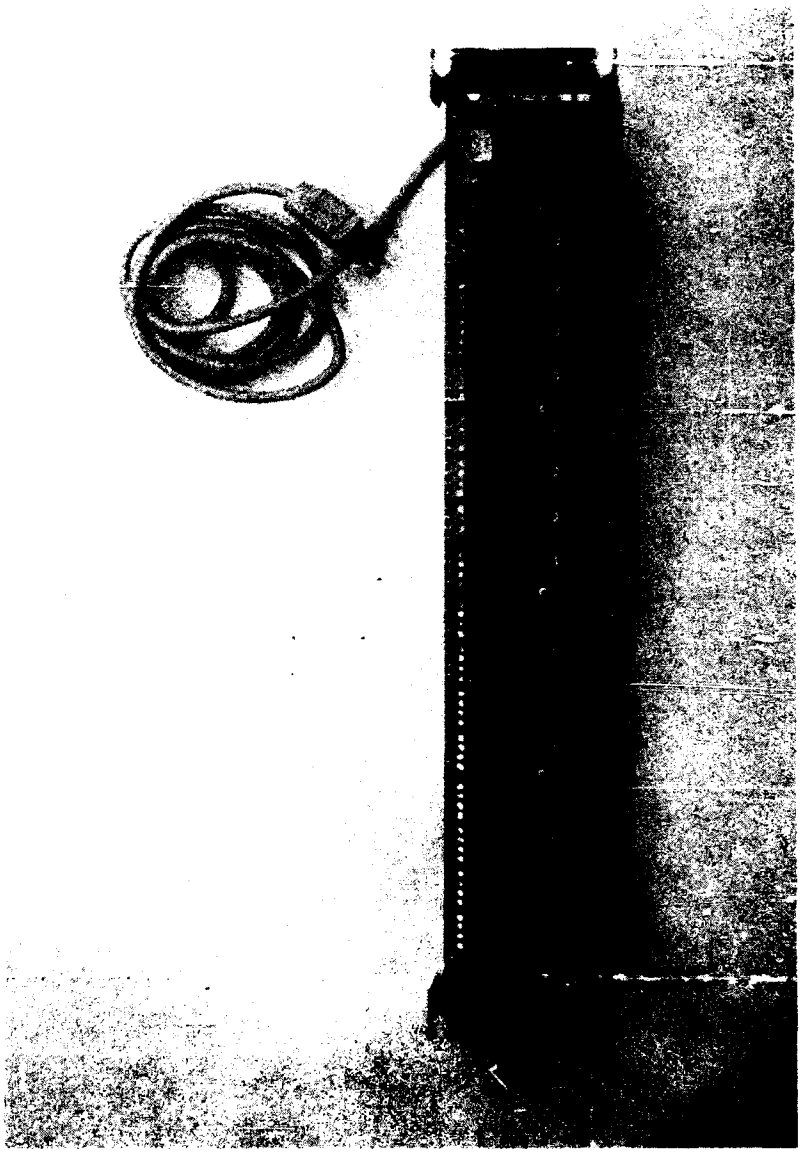


Figure 5 General purpose (GP) board

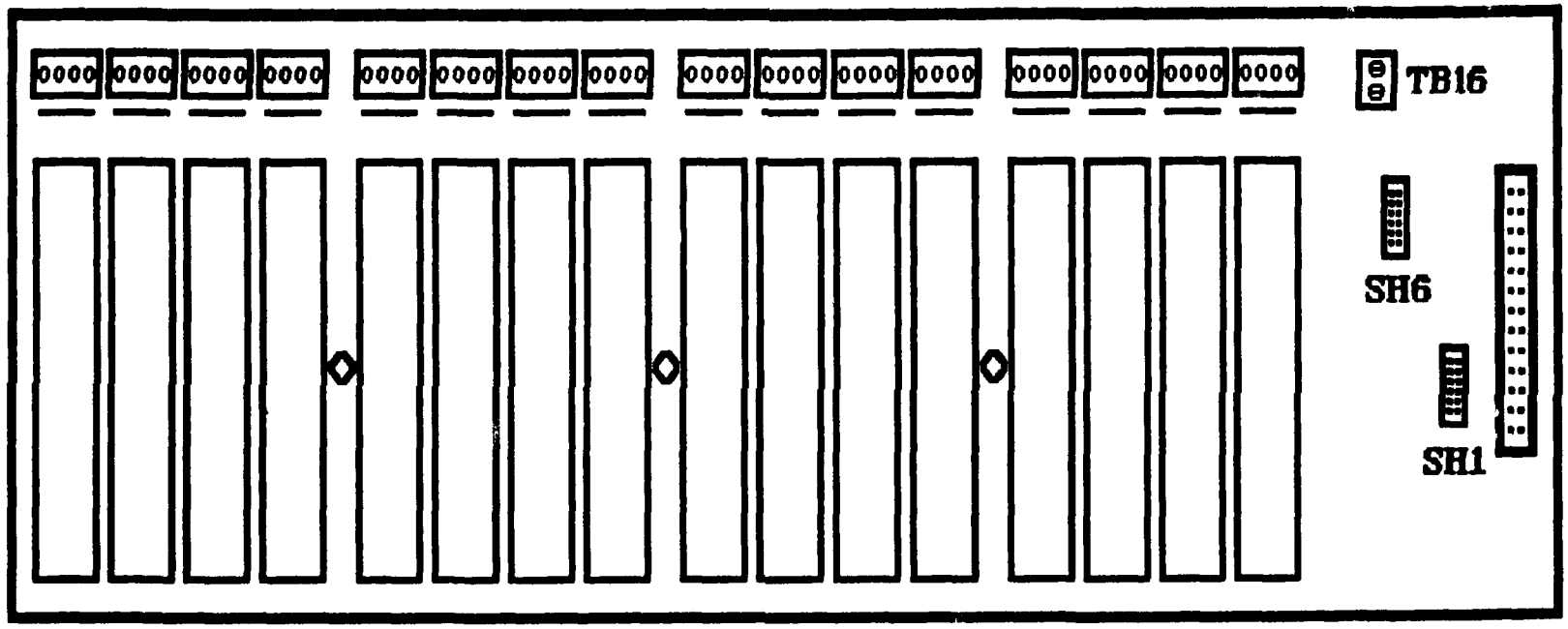


Figure 6 General purpose (GP) board layout

Thermistor Board

Figure 7 shows the thermistor (TM) board (MetraByte EXP-RES). The main function of the TM board is to convert the thermistor resistance measurement to an analog voltage value which the A/D board can accept and process. The four TM boards are mapped to addresses 1, 2, 3, and 4 of the A/D board input address. The A/D board map address is selected on the TM board by placing a jumper in the appropriately marked place in the OUTPUT CHAN jumper block (shown as OUT CHAN on the TM board layout schematic diagram of Figure 8).

The four TM boards can be cascaded together even though each has its own individual A/D board map address. The two parallel, 37-pin, connectors on the TM boards (see Figure 8) are identical in function and connections, and are used to cascade the TM boards, with 37-wire flat ribbon cables (MetraByte C-1800).

Each of the TM boards is equipped with its own +5V power supply. The +5V and Gnd supplies are connected through the connection block at KX1 (see Figure 8).

More detailed information about the TM board can be found in the MetraByte Users Guide for the EXP-RES Signal Conditioning Multiplexer Module.

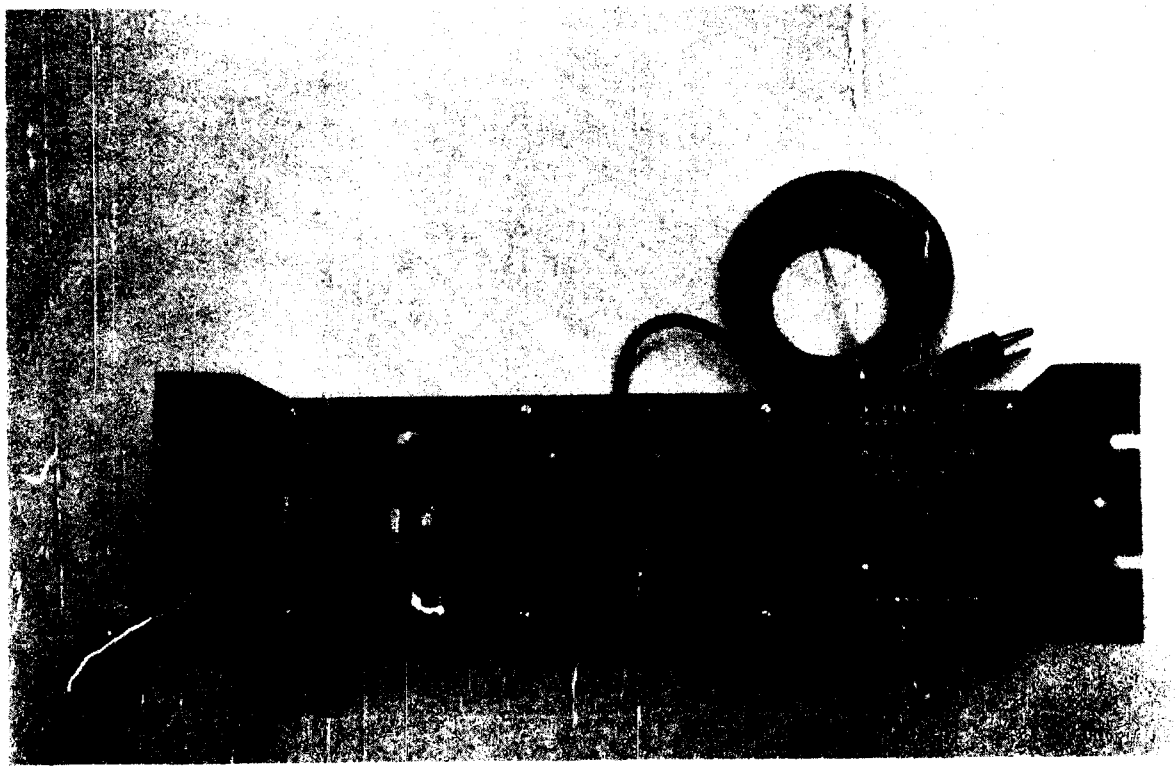


Figure 7 Thermistor (TM) board

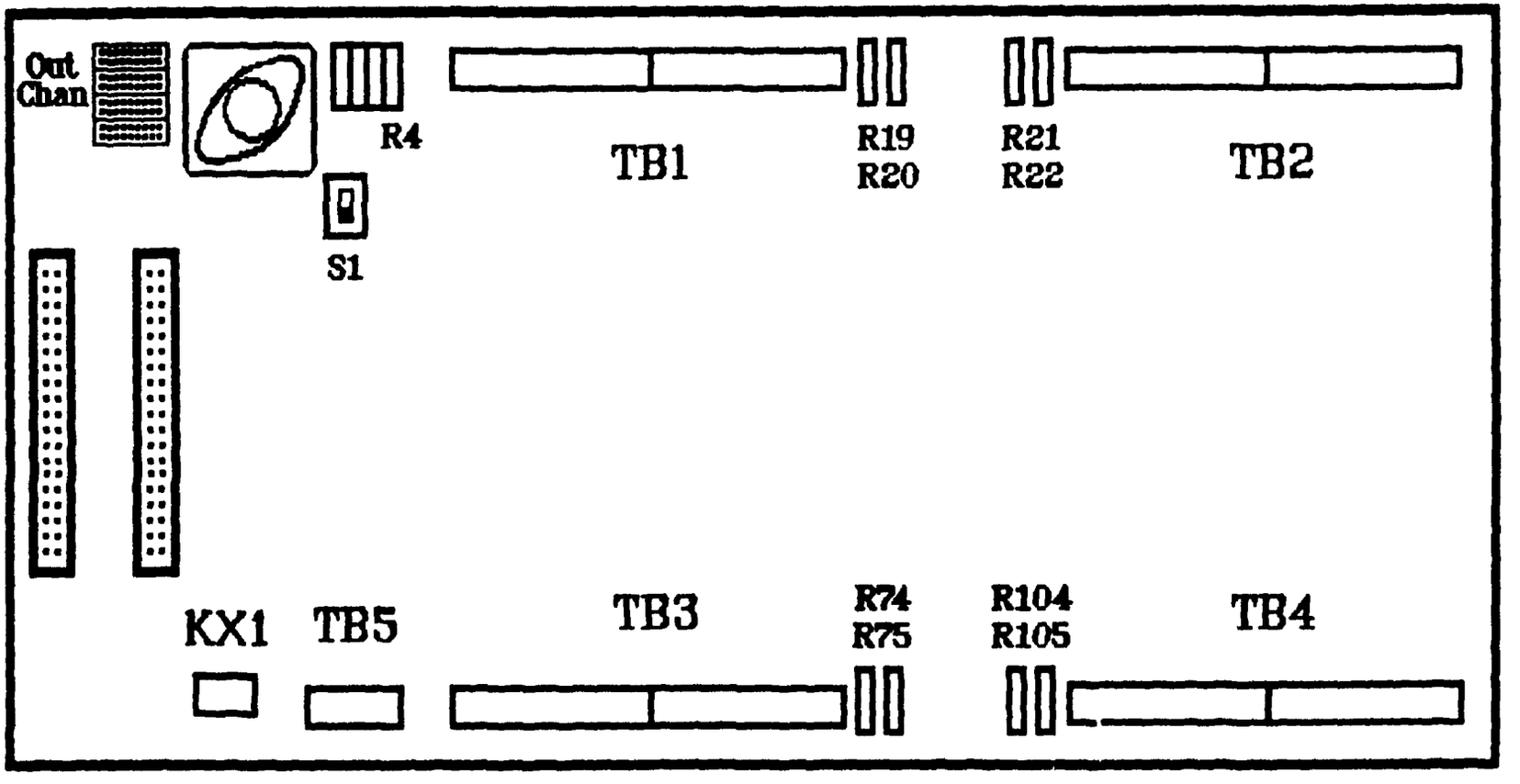


Figure 6 Thermister (T.) board layout

Inter-Connection Board

The GP boards and TM boards are brought together through two inter-connection (InC) boards (see Figure 9). Each InC board is a MetraByte STA-20. The dark bands in the schematic board layout (Figure 10) represent either 26-wire or 50-wire flat ribbon cables, described below.

As previously described in the GP board section, the two GP board cascade series are mapped to A/D board addresses 14 and 15. The GP cascade map addresses are selected by changing a specific connection on each of the two InC boards. One is hardwired to address 14, the other to address 15 (see Figure 10) Detailed information on the hardwired address change is presented in the Modifications section below.

The three cascade series (two GP, one TM boards) are brought together on the InC board in the manner shown in Figure 10. On each of the InC boards, there are 3 connectors. One connector has 26 pins (shown as the shorter connection in Figure 10), the other two are of the 50-pin variety (these two 50-pin connectors have identical function and connections). The two GP cascades are connected to the 26-pin socket, with 26-wire flat ribbon cables The thermistor cascade can be connected to either one of the 50-pin sockets. The two InC boards are then connected together using one 50-pin connector on each board. These connections use 50-wire flat ribbon cables.

The remaining 50-pin connector, shown as unused on the address-15 InC board in Figure 10, is for connection to the A/D board. The connection between InC and A/D boards also uses a 50-wire flat ribbon cable.

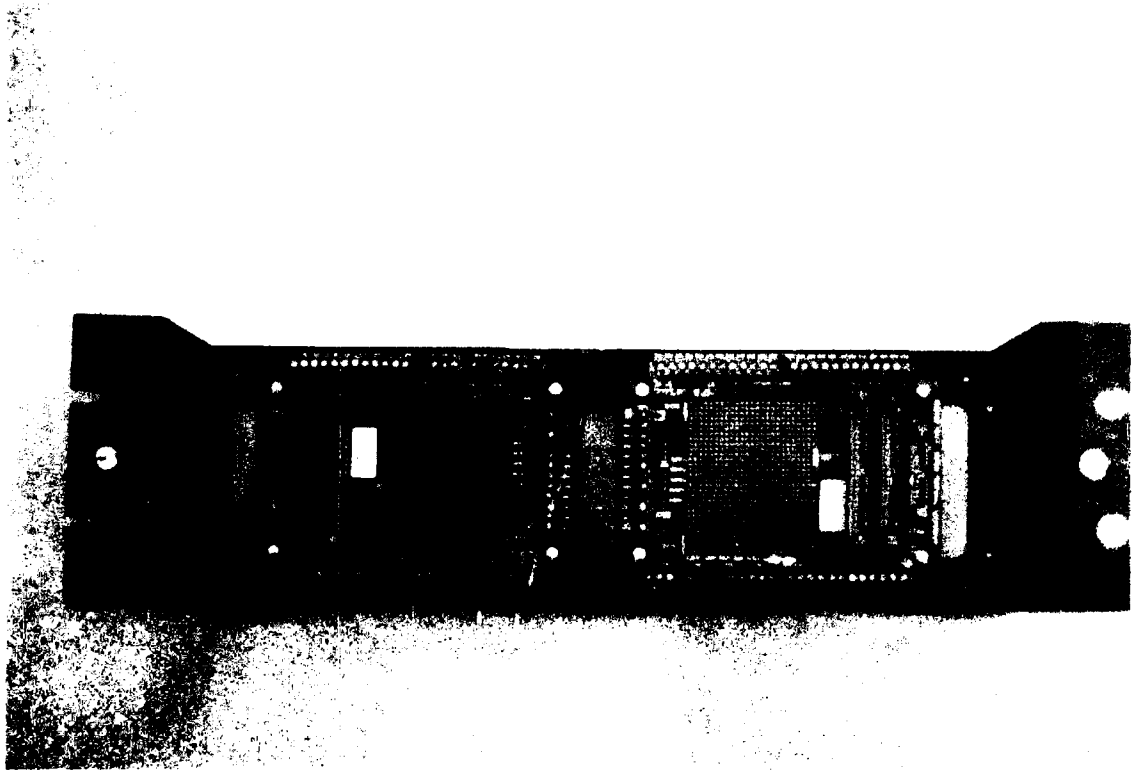


Figure 9 Inter-Connection (InC) boards

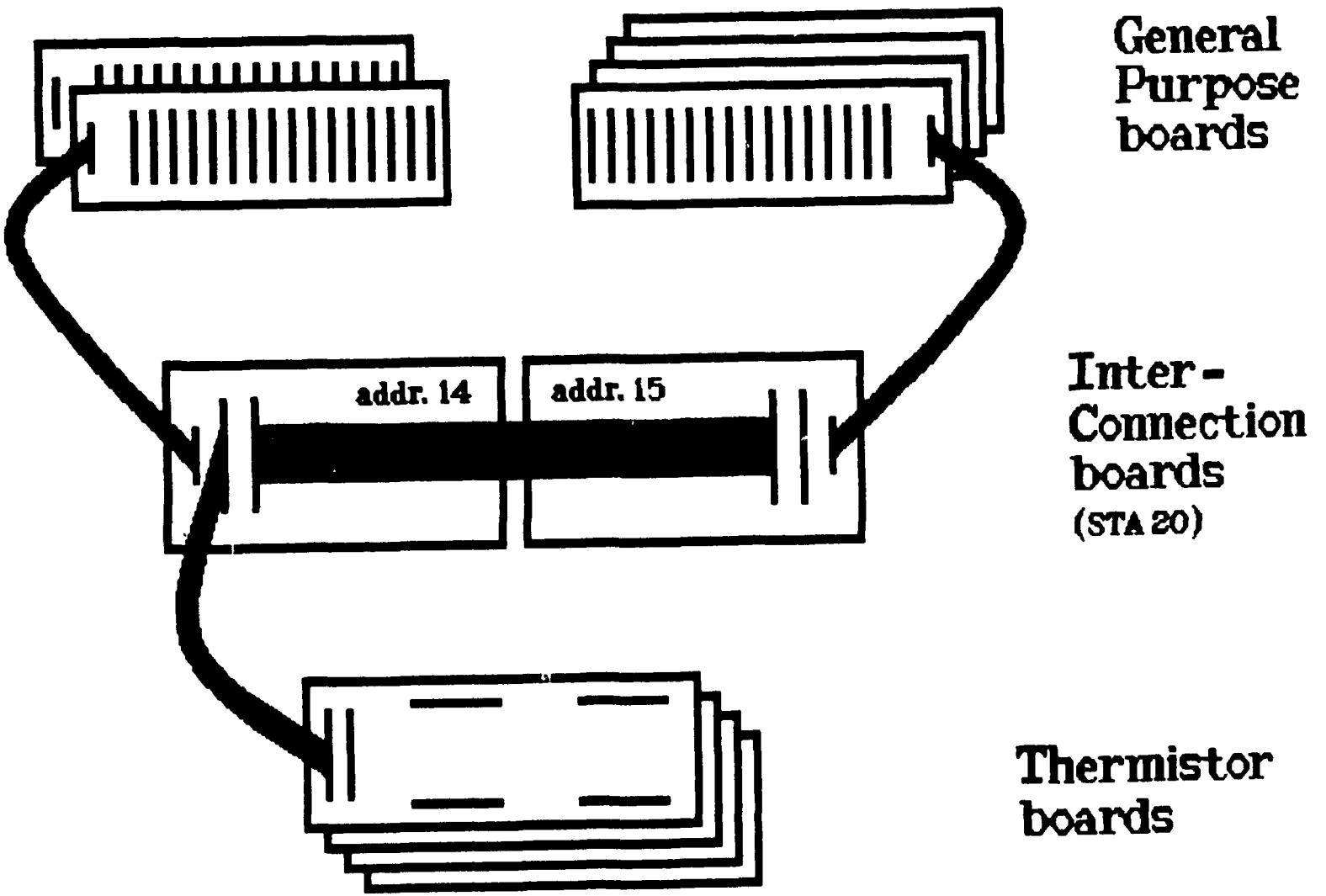


Figure 10 Inter-Connection (InC) boards and board connection schematic

INPUT TYPES AND MODULES DESCRIPTION

Analog Devices 5B Series input modules are used to accept the GP board inputs. Figure 11 gives a sample of the modules. All the available 5B Series modules can be used with the AC128. The input ranges and characteristics of each module are described below. More detailed information about the 5B modules can be found in the Analog Device 5B Series User's Manual.

Voltage

Voltage signals are collected using the 5B31 modules. There are three available 5B31 module types, each is dedicated to a specific input voltage range. The following selection list gives the specifications of the voltage modules.

input voltage range	module	resolution	output
± 1 V	5B31-01	0.488 mV	± 5 V
± 5 V	5B31-02	2.44 mV	± 5 V
± 10 V	5B31-03	4.88 mV	± 5 V

The potential input voltage range determines the type of module to be used. The proper module should be the one with the smallest input range, yet still able to contain the highest (or lowest if input is a negative voltage) possible input signal from the input channel. This will guarantee the highest resolution. Resolution is defined as the smallest voltage difference (in millivolts, $mV=10^{-3}V$) that each module can accurately and reliably differentiate. The 5B31 modules scale the input linearly to a corresponding output between -5V and +5V. The 5B31 modules have a bandwidth (-3dB point) of 4Hz (Hz=cycle/sec). The input signals are connected to the module through the input terminal blocks. Figure 12 (left side) depicts the GP board input terminal block. The positive (higher potential) lead should be connected to the Hi (no. 3) terminal. The negative (lower potential) lead is connected to the Lo (no. 2) terminal.

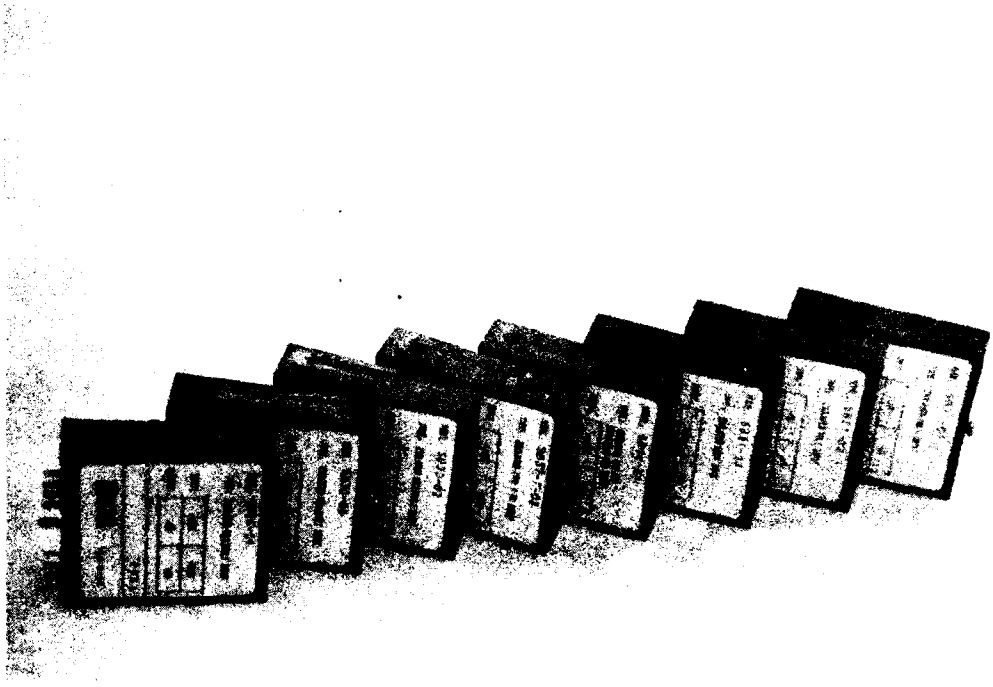
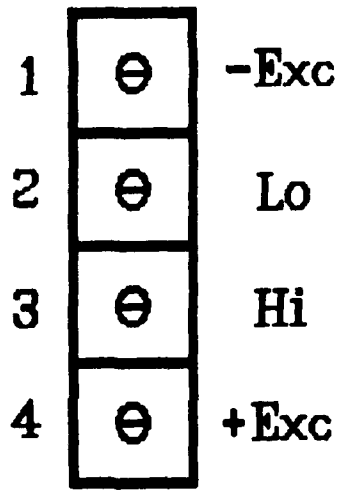
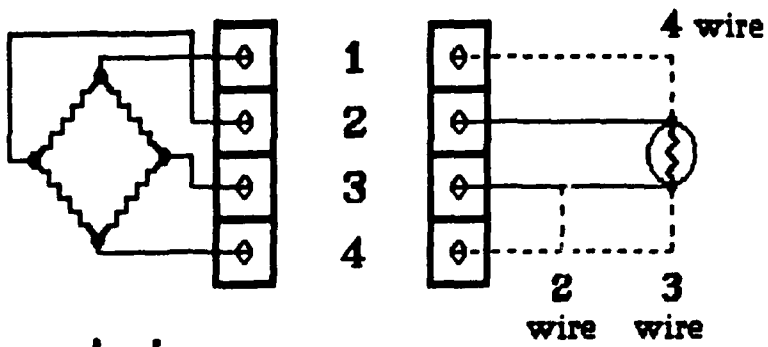
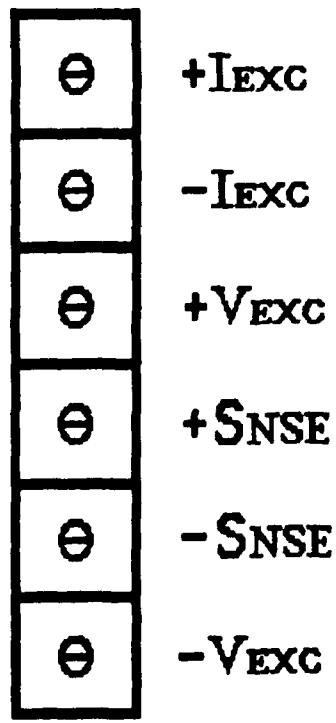


Figure 11 Sample of GP board input modules



on General Purpose board





on Thermistor board

Figure 12 GP board and TI board terminal blocks

Millivoltage

The millivoltage modules are designated as 5B30. Specifications of the 5B30 modules are contained in the following selection list.

input millivoltage range	module	resolution	output
± 10 mV	5B30-01	4.88 μV	± 5 V
± 50 mV	5B30-02	24.4 μV	± 5 V
± 100 mV	5B30-03	48.8 μV	± 5 V

The potential input voltage range should determine the appropriate module type to be used. The resolutions are shown in microvolts (μV). The output is linearly scaled to ±5V. The bandwidth of 5B30 modules is 4Hz. Millivoltage modules also use the Hi/3 and Lo/2 connections on the input terminal block.

Wide Bandwidth Voltage and Millivoltage

The wide bandwidth voltage modules 5B41 and wide bandwidth millivoltage modules 5B40 differ from the 5B30 and 5B31 modules only in the bandwidth. 5B40 and 5B41 have a 10 kHz (kHz=1000Hz) bandwidth. The two module selection lists above are applicable, with only the 5B30 and 5B31 designations changed to 5B40 and 5B41, respectively. All other relevant technical specifications are the same between these two groups of modules

Thermocouple

Temperature is the most commonly acquired data variable in biophysical and physiological studies. Thermocouples (TC) are commonly used to gather temperature data. The thermocouple signals are collected using the 5B37 modules. Different modules are used with each standard TC type. The module names contain the TC type designation letters (J, K, T, E, R, S, B), for example, the type T module designation is 5B37-T-03. Standard TC types are designed for different temperature range applications. For the temperature ranges encountered in most physiological studies, type T is usually apporpriate. Input range of the type T module is from -100°C to +400°C. The TC modules have a bandwidth of 4Hz.

Advantages of the thermocouple are versatility, ease of use, and low cost. The disadvantage is the comparatively limited accuracy. For example, according to NBS monograph 125[1], the standard limits of error for the commercial type T Copper-Constantan thermocouples are ±0.8°C, between -59°C and +93°C. Some manufacturers may subscribe to a more stringent limit of error, and different TC types may have different accuracy limits.

TC also require a reference temperature. Each channel on the GP board is provided with a built-in electronic TC cold-junction reference. The 5B37 module automatically compensates the measured temperature based on the electronic reference. The accuracy of the cold-junction reference also affects the overall accuracy. A basic method for checking TC accuracy is to calibrate against an ice-water solution (crushed-ice and water solution in a Dewar container producing a eutectic medium at 0°C). For the AC128 system, it is reasonable to make allowance for thermocouple measured data to be within ±0.3°C of the actual temperature. The measured temperature is stable, and is reproducible with the same TC.

The 5B37 modules convert thermocouple voltage to a corresponding output from 0 to +5V. The conversion from the thermocouple voltage to 0 -- +5V is linear. However, the thermocouple voltage itself is inherently nonlinear with respect to temperature. The AC128 SoS uses the National Bureau of Standards (NBS) type T TC conversion equation to derive temperature data from the voltage data. The conversion procedure will be described in detail in the SOFTWARE section.

The TC module uses the Hi/3 and Lo/2 input terminal block configuration. For the type T thermocouple, the high potential copper lead should be connected to the Hi terminal, and the low potential constantan lead to the Lo terminal.

---

[1]U.S. Department of Commerce. Thermocouple Reference Tables Based on The IPTS68 - NBS Monograph 125, p.180. National Bureau of Standards, Washington D.C., 1974.

Current Input

The current input module 5B32 converts, linearly, a current input between 0 -- 20 mA (milliamperes) to an output voltage of 0 -- +5V, by reading the voltage across a precision 20Ω resistor (Ω=Ohm, basic unit of electrical resistance). The bandwidth is 4Hz. Input terminal block connection follows the Hi/3 and Lo/2 configuration.

RTD

The platinum RTD is used for precision temperature measurement. The 5B34-01 RTD module accepts 100Ω (α=0.00385) platinum RTD probe. The temperature range is between -100°C and +100°C. Bandwidth of the module is 4Hz. The input lead connections depend on whether the RTD is a 2, 3, or 4 wire device. The table below shows the connections. The special connection configurations for RTD are also included in Figure 12.

RTD type	Connection terminals used
2–wire	3, 4
3–wire	2, 3, 4
4–wire	1, 2, 3, 4

Strain Gauge

The 5B38-01 modules accept signals from a full bridge (100Ω to 10kΩ) strain gauge transducer. The module supplies an excitation of 3.33V, and produces an output from -5V to +5V. The 5B38 module has a wide bandwidth of 10kHz. The four leads of the strain gauge bridge are connected to the four terminals on the input terminal block. A schematic of the strain gauge connection configuration is also included in Figure 12.

Thermistor

Thermistors (TM) are handled specially by the thermistor boards. TM are also frequently used to measure temperature. The main advantage of TM is their sensitivity, i.e. a small change in temperature will invoke a large change in TM output signal. The nominal accuracy of a TM is ±0.1°C. Thermistors are temperature sensitive resistors, generally composed of semiconductor materials. Unlike TC whose natural output is a voltage, the TM output is a resistance. Most TM exhibit a negative temperature coefficient, i.e. their resistance decreases as the temperature increases. Resistance is not a readily recognizable signal to an electronic system such as the PC. The task of AC128 is thus to 1) transform the resistance signal to a voltage signal so that it is recognizable to the acquisition hardware; 2) collect the voltage data; and 3) convert it back to temperature data.

Recall from the above discussion, TM must be connected to channel 96 -- 127, i.e. must be connected to the TM boards. The AC128 allows from one to 32 TM to be used. If less than 32 TM are used, the unused channels do not need to be prepared specially, such as terminating with a load resistor. However, an unused TM channel should be disabled, as should all unused channels, to maximize data acquisition efficacy. (Channel disabling procedures will be discussed in the SOFTWARE section).

The TM temperature data transformation and conversion process demands both special hardware and software. Hardware is used to transform a resistance value into voltage data. An excitation voltage source is required, and a bridge circuit is employed for accurate measurement. The special bridge circuit used in AC128 is shown in Figure 13. The necessary hardware circuitry is incorporated and accommodated entirely on the TM board. The TM leads are connected to the -Snse and -Vexc positions on the TM board input terminal block (see Figure 12, right side). The polarity of the leads is not significant. From Figure 13, it can be seen that the -Snse and -Vexc leads connect the thermistor directly into the TM resistance measurement bridge circuit. The A/D board then measures the voltage across the +Snse and -Snse terminals.

Following the hardware conversion (from resistance to voltage), software is used to convert the voltage signal back to temperature data. Detailed discussion of the conversion equations will be discussed in the SOFTWARE section.

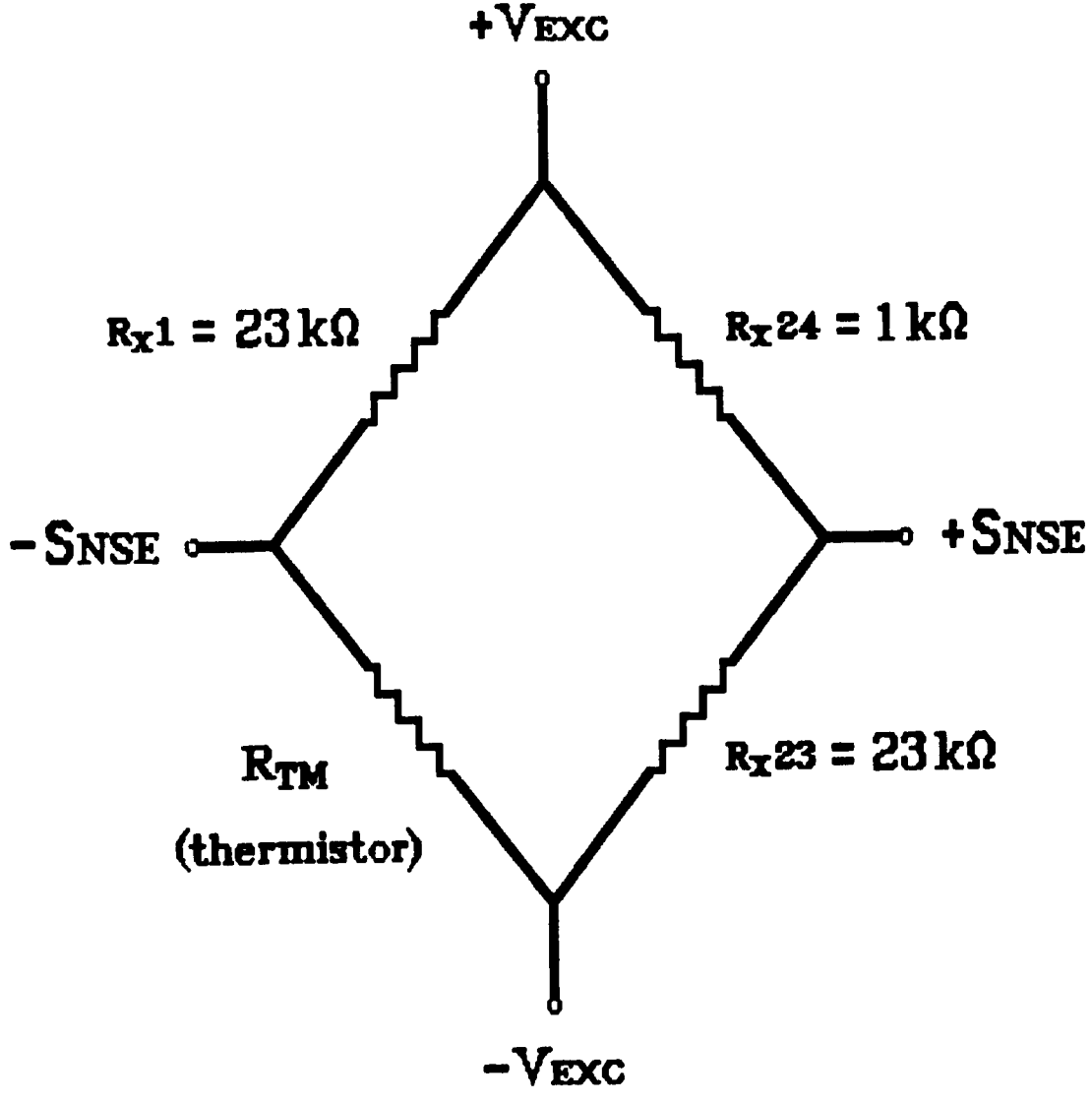


Figure 13 Thermistor board bridge circuit shematic

HARDWARE MODIFICATIONS

Inter-Connection Board

One modification is required on each of the two InC boards. The InC boards were factory prewired to an address corresponding to A/D board address 0. Since we require these boards to be mapped to A/D board addresses 14 and 15, therefore the connection to address 0 must be cut and new connection to address 14/15 must be hardwired. On the back side of the boards, the connection between pin 1 of the 26-pin connector and pin 1 of the 50-pin connector is cut, and new connection made between pin 1 of the 26-pin connector and pin 14/16 (for address 14/15, respectively) of one of the 50-pin connector. The new connections were made using green wires which are visible on the back side of the InC boards.

Thermistor Board

The TM boards require several modifications. On the TM boards, the 16 resistors in positions R114 through R129 must be 0Ω. On early versions of the thermistor board, 0Ω resistors were used. In later versions, the board manufacturer has substituted nonzero ohm resistors in these 16 positions. The nonzero ohm resistors must be removed and replaced with 0Ω resistors, or simply straight wires. These changes are clearly evident as the resistors have been replaced with gold plated wires on all of the TM boards.

Other modifications on the TM board are necessary to make the bridge circuit for thermistor resistance measurement. The schematic of the bridge circuit in Figure 13 has already been discussed. On the board, a precision (0.01% tolerance) 1kΩ (kΩ=1000Ω) resistor was placed in the RX24 position. For the RX1, RX2, RX23, RX26, RX27, RX48, RX49, RX50, and RX51 positions, a precision 23kΩ (0.01% tolerance) resistor was placed in each. Also, a connection must be made from the +Vexc position on TB5 to one of the +Vexc positions on TB3. -Vexc of TB5 must be connected to -Vexc on TB3. All +Snse positions on TB1 through TB5 must be wired together. These connections were added using yellow wires on the back side of the TM boards.

On the front side of each TM board, a two-connection terminal block was removed from TB5 and placed in the KX1 position to facilitate the power supply connection. The KX1 terminal block now accepts the +5V and Gnd from the power supply. On the back of the TM boards, the +5V and Gnd from KX1 are connected to pin 29 (green wire) and pin 11 (black wire), respectively, of one of the 37-pin connectors on the board.

CALIBRATIONS AND MAINTENANCE

For the AC128 hardware component boards shown in Figure 1, the GP boards and the InC boards do not require periodic calibration. The A/D board and the TM boards may be calibrated. To perform the described calibrations, the following equipment and tool are required:

$4^1/_2$-digit digital voltmeter
low noise voltage source (adjustable over ±5V range)
small slot-head screwdriver or potentiometer trimmer.

Analog-to-Digital Board

The A/D board calibration should not be necessary unless there is strong indication of error, such as an undeniable erronecus voltage reading. If an A/D board calibration is necessary, it should be performed prior to the TM board calibration. The calibration of the A/D board must be performed outside of the AC128 SoS environment (i.e. SoS cannot be used). Also the calibration must be performed with the PC cover removed. Refer to Figure 4 for potentiometer designations and approximate positions on the A/D board.

1. Turn PC power off.
2. Remove the PC cover.
3. Disconnect the flat ribbon cable between the InC and the A/D boards.
4. Carefully remove the A/D board from the expansion slot.
5. Reconfigure the A/D board for differential input mode by setting jumpers W9B to W9C, W8B to W8C, and W1B to W1C on the A/D board.
6. Reinsert A/D board back into the expansion slot.
7. Connect the low noise voltage source to channel 0. On the 50-pin connector from the A/D board, connect pins 1, 2, and 17 to the voltage source's high, low and ground leads, respectively. These 3 pin positions represent channel 0 high, channel 0 low, and input sense of the A/D board (J2 connector).
8. Turn PC power on.
9. Set the voltage source to -5.0 volt.
10. Write a short program to sample channel 0 for input range ±5V, gain setting of 1, and display sampled voltage reading on the PC monitor.
11. Adjust potentiometer R19 on the A/D board until the PC monitor displays -5.0 volt.
12. Now, set the voltage source to +5.0 volt.
13. Use the sample short program to sample channel 0.
14. Adjust potentiometer R18 until the PC monitor displays +5.0 volt.
15. Turn PC power off.

16. Carefully remove the A/D board from the expansion slot.
17. Configure the A/D board for single-ended input by setting jumpers W9A to W9B, W8A to W8B, W8C to W8D, and W1A to W1B.
18. Reinsert the A/D board back into the expansion slot.
19. Replace the PC cover.
20. Turn the PC power on.
21. Record the date of calibration.

More detailed information about the A/D board calibration procedure can be found in the Analog Device RTI-815 User's Manual.

Thermistor Board

It is advisable to calibrate the TM boards every 12 months. TM board calibration (refer to Figure 8 for terminal and connection designations) can be performed within the AC128 SoS. Two types of periodic calibrations are necessary on the TM boards. The first calibration is for the thermistor resistance measurement bridge circuit excitation voltage.

1. Make sure switch S1 is set to the +5V position.
2. Connect the +Vexc and -Vexc of TB5 to the high and low leads, respectively, of the voltmeter.
3. Adjust potentiometer R4, by turning clockwise or counterclockwise, until the voltmeter reads 4.0 volts.

Each thermistor channel on the TM board must also be zero adjusted. These TM channels must be calibrated one at a time.

1. Use the AC128 SoS to select one and only one TM channel for data collection.
2. On the terminal block of the selected channel, wire together the +Snse, -Snse, and -Iexc positions.
3. Connect the low lead of the voltmeter to the -Iexc position on any one of the TB, and connect the voltmeter's high lead to one of the jumper pins on the lower side of the OUTPUT CHAN jumper block (shown as OUT CHAN in Figure 8).

4. Adjust the appropriate potentiometer, from table below, until the voltmeter reads zero.

TM channel	Potentiometer
0	R19
1	R20
2	R21
3	R22
4	R74
5	R75
6	R104
7	R105

5. Repeat steps 1 through 4 for each thermistor channel on each TM board.
6. Record the date of calibration.

More detailed information about the TM board calibration procedure can be found in the MetraByte Users Guide for the EXP-RES Signal Conditioning Multiplexer Module.

SOFTWARE

The AC128 software system (SoS) was written by the author using primarily QuickBASIC, version 4.0, with small interface and data collection routines in Assembly language. This section should also serve as an operations manual to guide the operator through the SoS.

The SoS adheres to a modular design. Each module may be subdivided into several submodules. The entire software program listing is included in this report as Appendix A. The program contains extensive comments to aid the understanding of the algorithms. A line-by-line description of the software will not be attempted. However, the module and submodule level algorithms are described in detail.

Figure 14 is a system level flow chart of the AC128 SoS. It is immediately clear that the SoS consists of four main modules, Initial Setup, Main Selection, Channel Setup, and Data Acquisition. Each main module may then be divided into submodules. Figure 15 details the submodules within Main Selection. The five primary channel setup submodules within the Channel Selection module are shown in Figure 16. Figure 17 gives further detail to two of the more complex submodules within the Channel Select module. Figure 18 shows the data acquisition submodules. Figure 19 is a combined overall system flow chart. Each module and the submodules within will be described in detail. In the following description, the names that appear between < > are label or subprogram names used in the SoS programs.

If the SoS is run on a PC with hard disk system, it is advisable that a separate directory be setup for the AC128 files. Initially, the directory should contain the executable system file AC128.EXE and the canned demonstration setup file DEMOS.S28. Other setup files tailored to specific studies will be added gradually to the directory. A backup of the AC128.EXE and setup files should also be maintained.

To start the AC128 SoS:

(1) Power up the PC.

(2) Change to the data acquisition directory (if separate directory is setup).

(3) At the DOS prompt, enter "ac128" (without the quotation marks), and then depress the ↵ Enter key. The SoS will initialize the system <NewSetup>. A version and date message is displayed (Figure 20). The latest AC128 version at the time of this report preparation is 2.0 (displayed as AC128.20). The date indicates the date of the last update. Also included in the initial display is a reminder that HELP screen(s) are available. The version and date message is displayed only once, at this initial startup time.

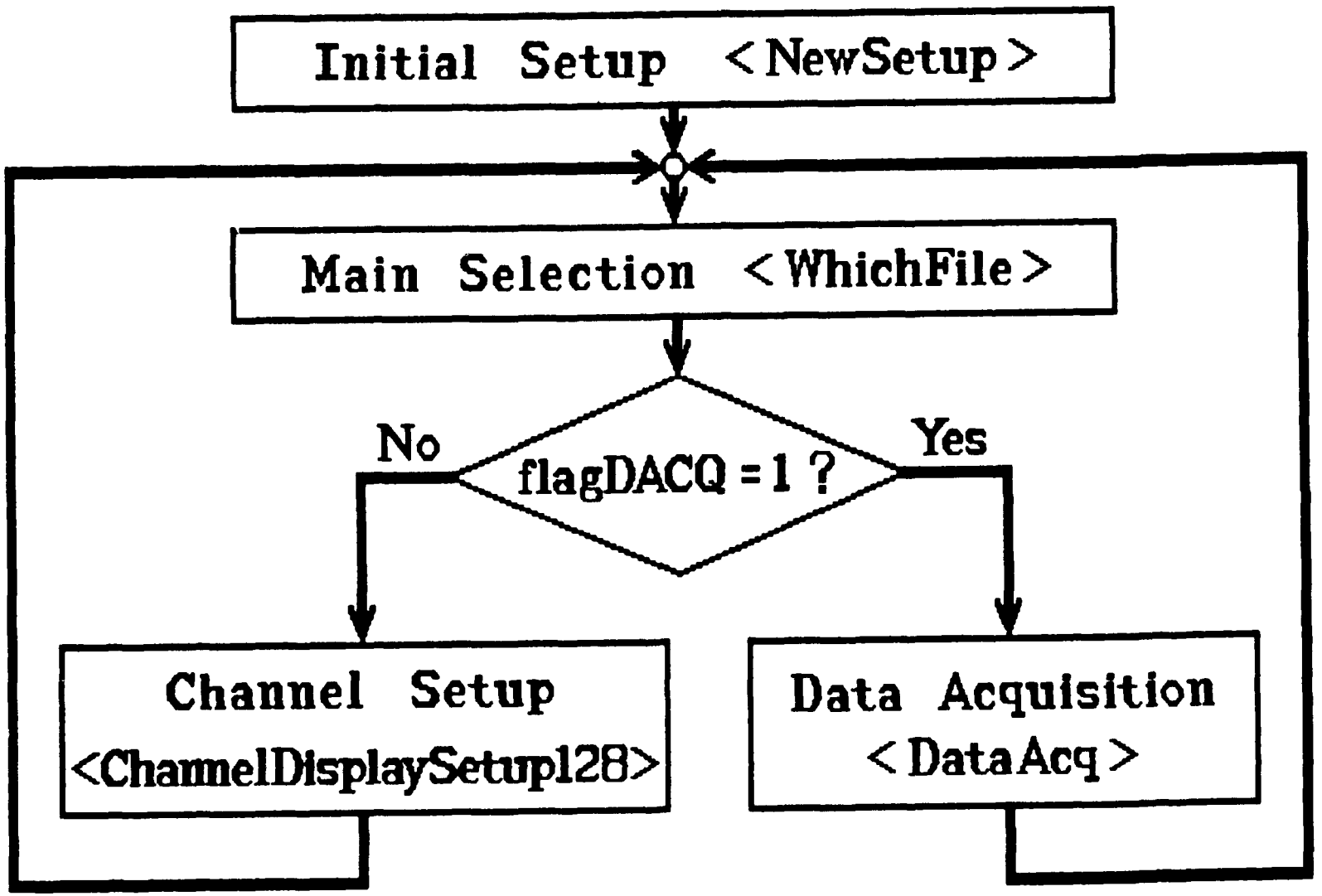


Figure 14 AC128 SoS system level flow chart

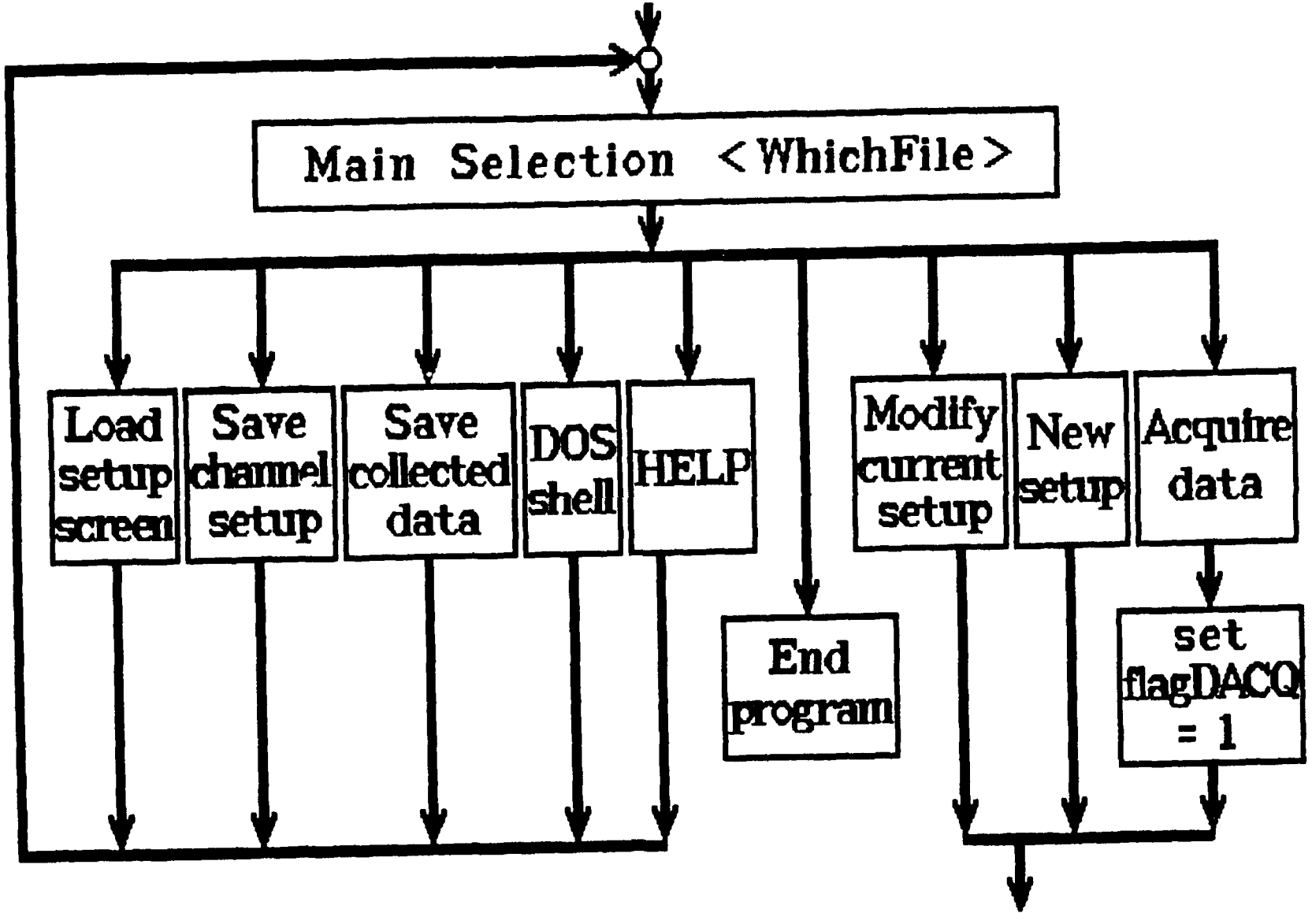


Figure 15 Main Selection module flow chart

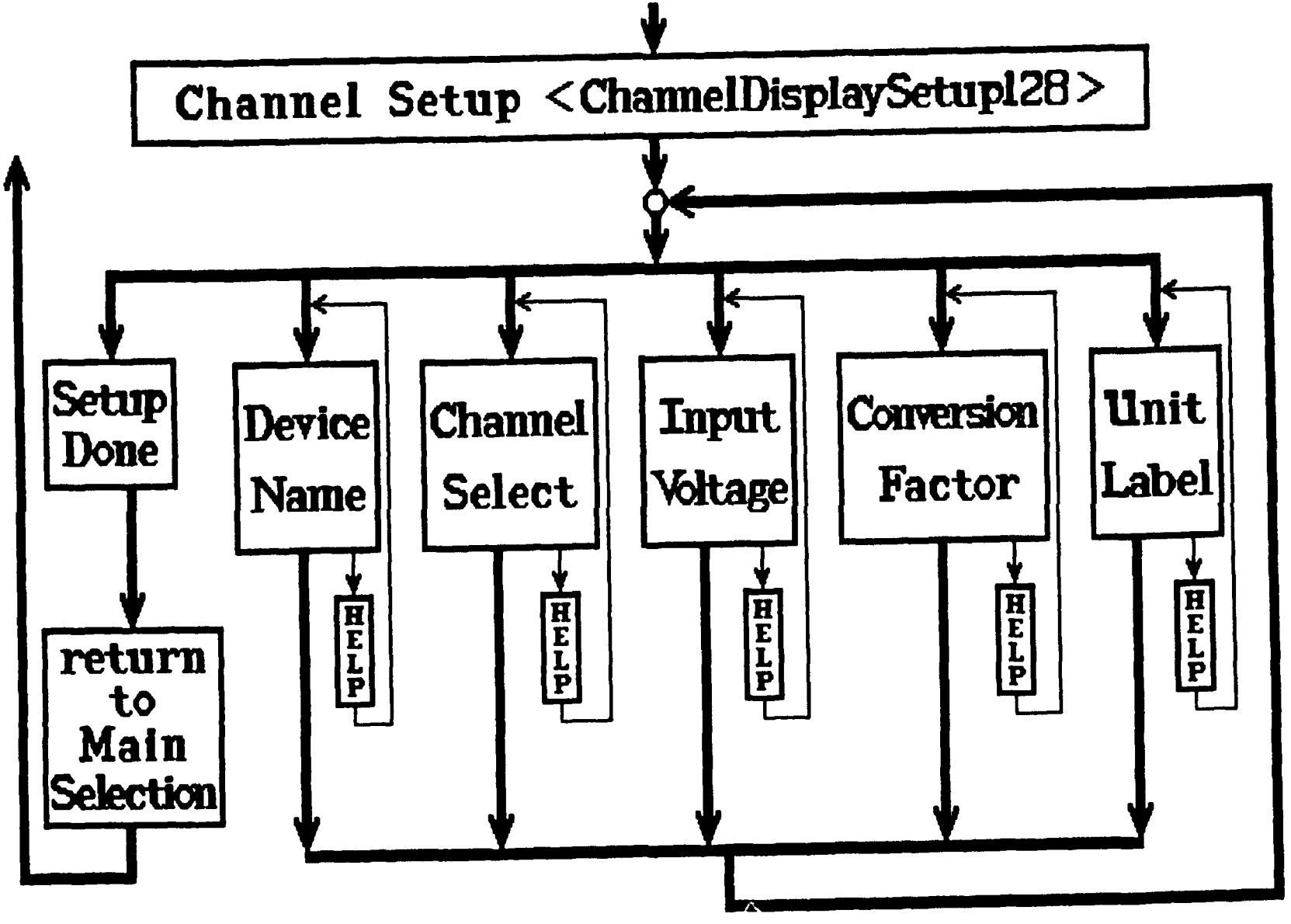


Figure 16 Channel Setup module flow chart

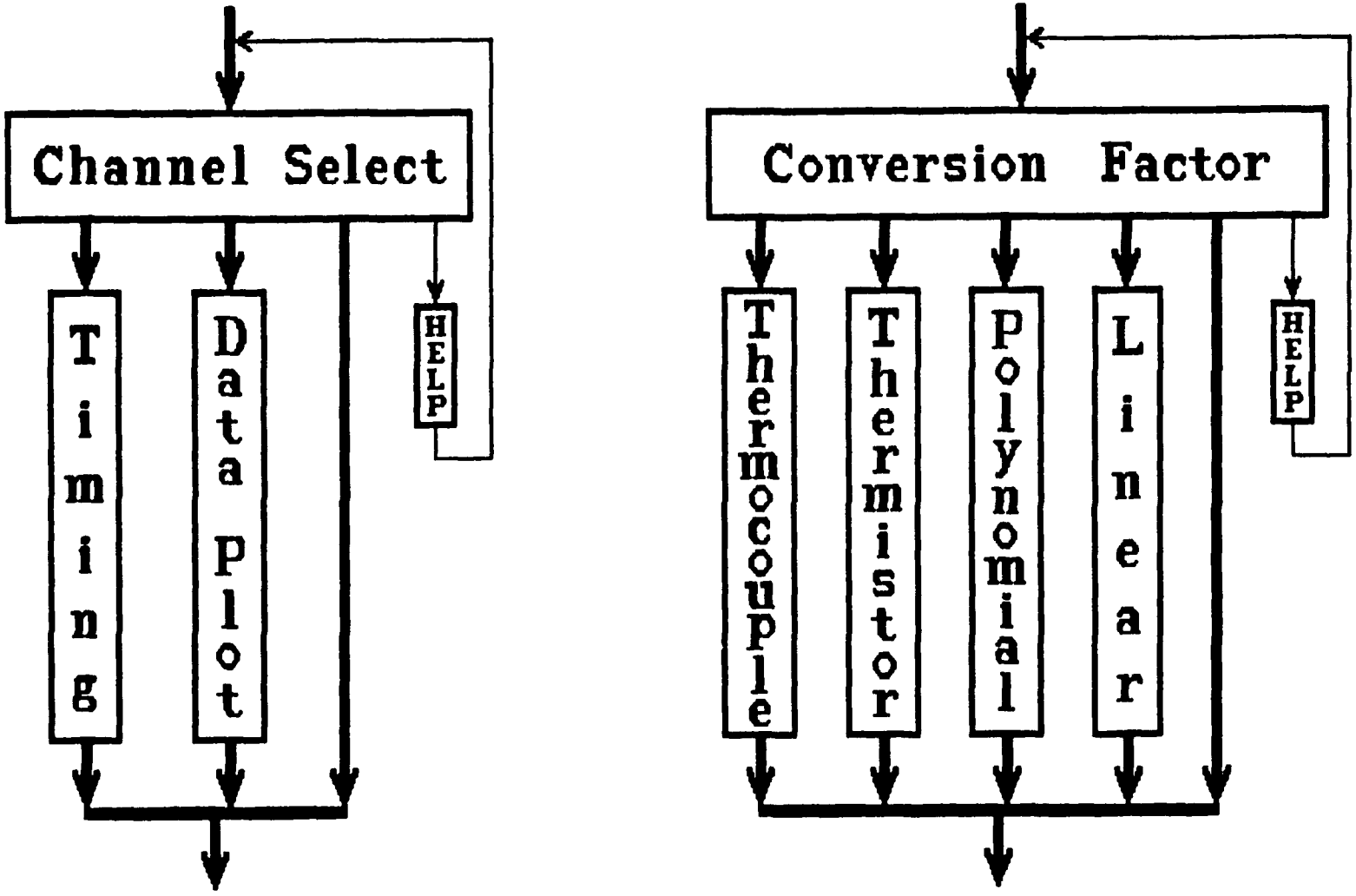


Figure 17 Detailed flow chart of Channel Select and Conversion Factor submodules

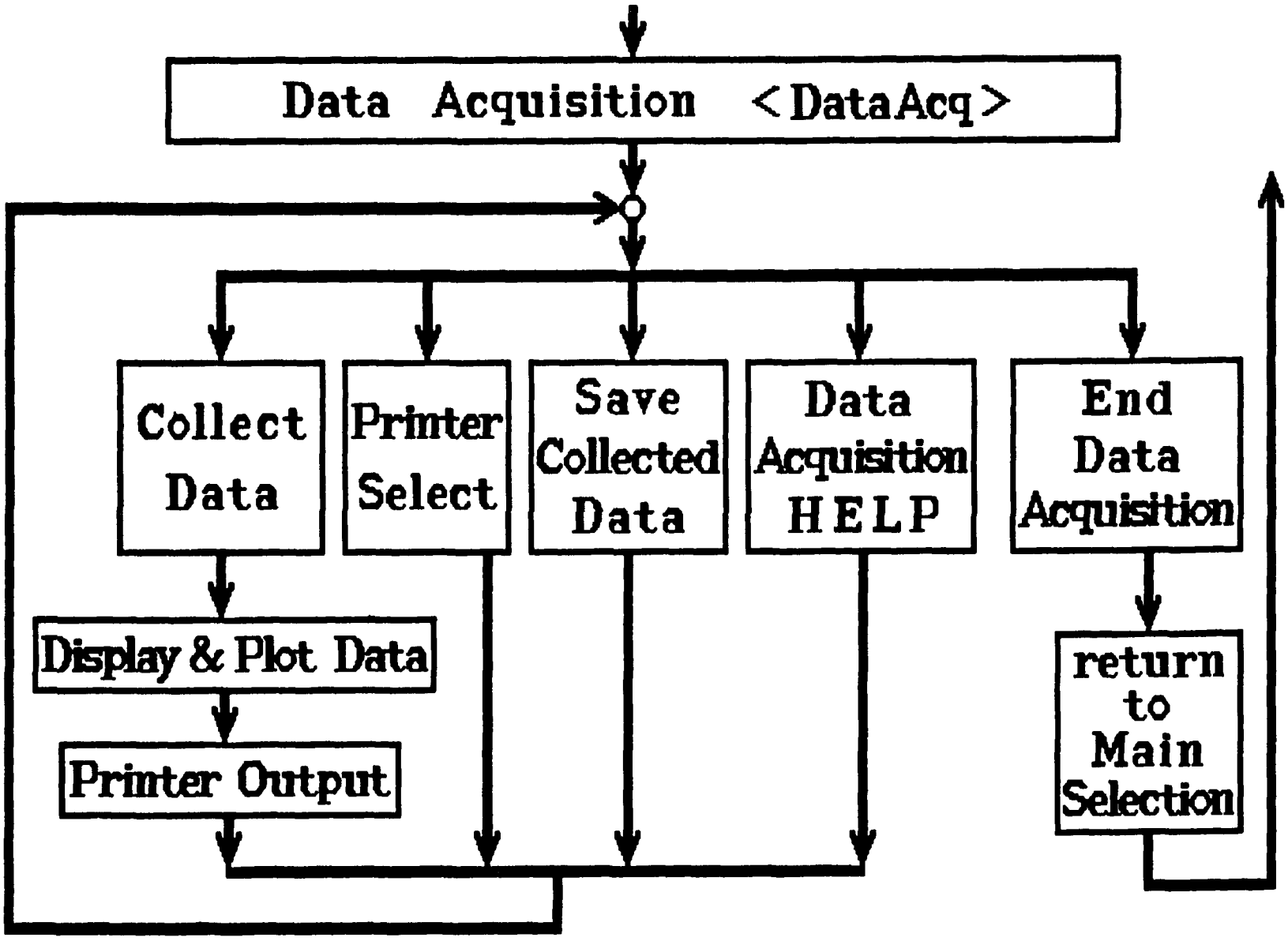


Figure 18 Data Acquisition module flow chart

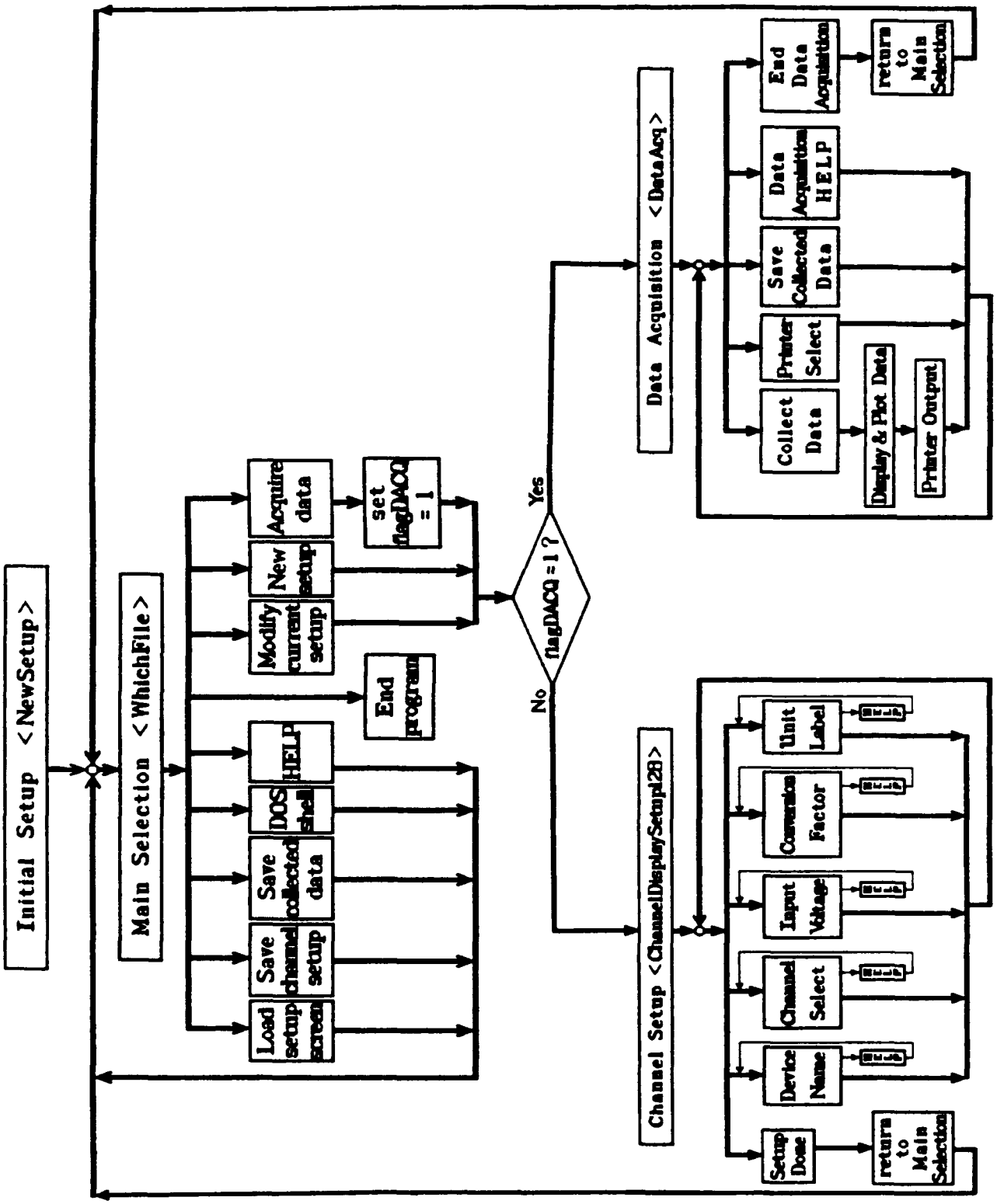


Figure 19 Combined overall AC128 software system flow chart

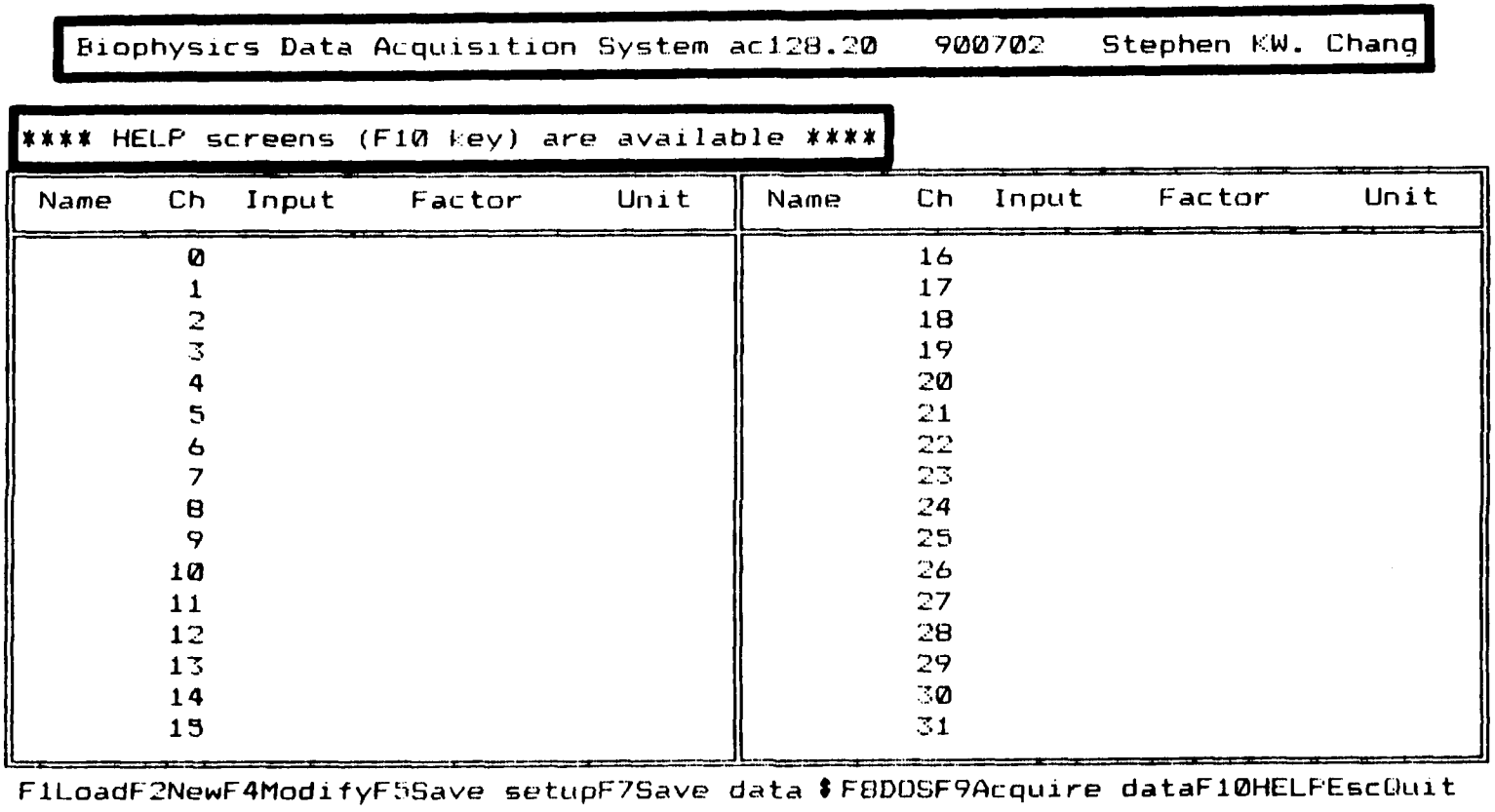


Figure 20 Initial AC128 SoS screen with version and date message, and HELP reminder

INITIAL SETUP <NewSetup>

The NewSetup module initializes the operating system. All the channel setup information is nulled. All operating system flags are cleared, and all variables are set to predefined initial values. The PC monitor screen is cleared and prepared for channel setup information display <DataScreen>. The four plotting screens are also initialized <PlotScreen>.

The monitor display screen is divided into four areas (Figure 21). Area (1) is the Prompt screen. System status, program warnings, error messages, and some channel setup information are displayed in this area. At the initial startup, no setup file has yet been loaded or named, hence the current setup file name in the Prompt screen is blank.

Area (2) contains the data acquisition timing information: data channel sampling interval, printer output cycle, and study period. The sampling interval and printer output cycle are in seconds, whereas the study period is in minutes. At the Initial Setup, all are defaulted to zero.

Area (3) is the main body of the display screen. Data acquisition channel setup information is displayed here. There are two halves to Area (3), each half is divided into five columns: Device Name, Channel Number, Input Voltage Range, Conversion Factor, and Unit Label. These five column contain data to be entered by the operator to setup the AC128 system for data acquisition. Device Name is a name label that the operator can enter to identify each channel. Channel Number (displayed as Ch) ranges from 0 to 127. Input Voltage Range can be ±5V, ±500mV, ±50mV, or ±10mV. The SoS defaults all input channel to the ±5V range, because most of the standard AC128 inputs have been configured to this range. Conversion Factor specifies how the raw input data should be processed to arrived at the meaningful unit, shown in Unit Label. These five column entries will be explained in more detail at the appropriate submodule level.

Notice each half of Area (3) displays 16 channels of setup information. The entire screen has room for 32 channels. Since the AC128 can accommodate 128 channels, therefore there are actually four display screens (pages), each with 32 channels of setup information. The initial display pages are identical, except for the channel numbers. Page 1, 2, 3 and 4 have channels 0 - 31, 32 - 63, 64 -95, and 96 - 127, respectively. The page numbers are shown in the upper left-hand corner of the Prompt screen (Figure 21). Display pages 5 through 8 are plotting screens which are used during data acquisition. The acquired data from a total of 4 channels may be plotted to give a time-course representation of the data. The plotting screens will be discussed at the appropriate submodule level.

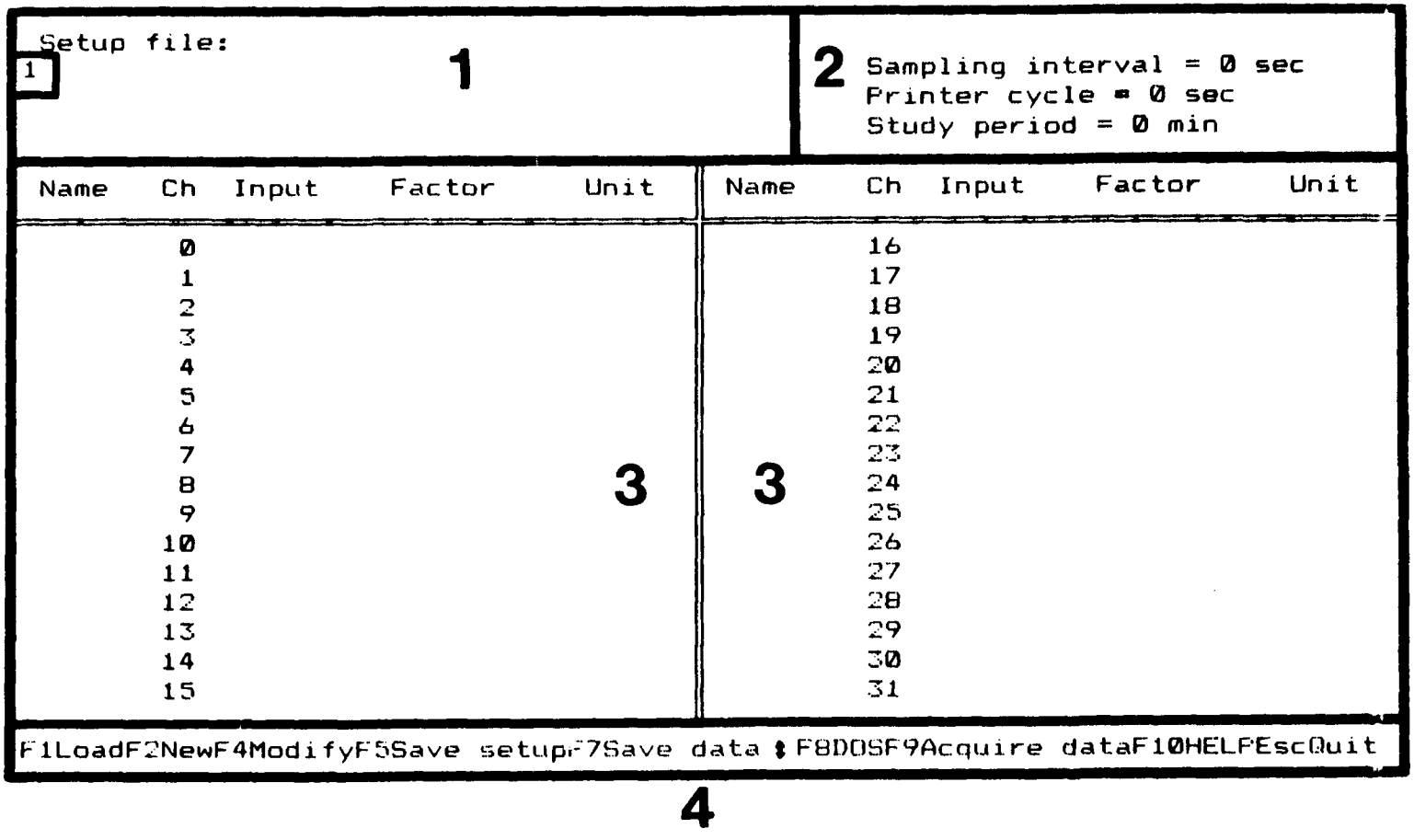


Figure 21 Four areas of display screen

Area (4) is the function key (F-key) row. This displays the function keys that are currently activated. The pressing of an activated function key selects a particular action to be performed. The action called by each function key depends on the stage wherein the keys are depressed. A stage refers to a module or submodule within the SoS. The one function key that has a constant definition throughout the SoS is the F10 key. F10 always bring up the HELP screen(s), which briefly explain what options are currently available to the operator. The actions of other F-keys will be explained later at the appropriate module level. There are other keys that may or may not be displayed in the F-key row but are usually activated. These keys also point to constant functions. PageUp (PgUp), and PageDown (PgDn) bring up another display page, either the previous or next 32 channels of setup information, or another plotting screen. In the Channel Setup module, the ↑, ↓, →, ← arrows are used to move the current active channel up or down, and between the ten columns (5 in each half) in Area (3). Another very helpful key is the ESC key. The ESC key always exits to the previous stage in the SoS. The ESC provides a quick way to cancel any unwarranted actions.

When the SoS opens a dialog box for data entry, the F-key row then shows a set of data entry functions. The following lists the possible functions.

BackSpace - deletes the character before the cursor.
Delete - deletes the character at the cursor position.
→ ← - moves the cursor right or left.
↓ - moves to the next entry, if more than one is asked by the SoS as in the coefficient set of a conversion factor.

↵ or Enter - signals to the SoS that data entry is complete.

F5 replace - brings up the replacement window. The data in current entry are replaced by corresponding data from the indicated replacement channel.

F9 keep - keeps all data entered thus far and retains all other unchanged data entries as they were.

Insert/typeover - switches between insert and typeover mode. In the insert mode, the cursor appears as an underline. In the typeover mode, the cursor covers covers the entire character.

Esc discard - the Escape key discards all entries entered thus far, and closes the dialog box.

MAIN SELECTION <WhichFile>

After the initial setup, the SoS proceeds into the Main Selection module. Figure 15 shows the submodule options available within the Main Selection stage. It should be immediately clear that the submodules are divided into two main groups. From the first group, the SoS returns to the Main Selection stage after the submodule function is completed. This group includes Load Setup Screen, Save Channel Setup, Save Collected Data, DOS Shell and Main Selection HELP screens. The second group comprises of Modify Current Setup, New Setup, and Acquire Data submodules. For this second group, at the completion of the submodule function, the SoS may be routed to either of two modules. If the Acquire data option is selected, a status flag <DACQ> is set to 1. A set <DACQ> flag directs the SoS to the Data Acquisition module, while a cleared <DACQ> flag routes the SoS to the Channel Setup module.

The function keys that activate the submodules are displayed in the F-key row in Figure 21. The 128 channels of setup information that are currently displayed on screen (only 32 channels displayed at a time) will be referred to as the active setup file.

Load Setup Screen <LoadSetup> F1–Load

Load previously saved setup file. The channel-by-channel setup for data acquisition can sometimes be quite time consuming, even though procedures have been included to simplify the process as much as possible. Once setup information is entered, it may be saved. The saved setup information can then be recalled for future uses. The setup file to be loaded back must conform to specified format, otherwise an error message will be generated. To be certain, the file to be loaded back should be one that was previously saved by the SoS.

The operator will be prompted with extensive information about the setup file format (Figure 22). If an error did occur, the operator will be asked to retry, with suggestions of possible sources of error (Figure 23). As an aid to the operator, a demonstration setup file "demos" has been included. The demonstration setup file may be loaded initially to allow a new operator to familiarize with the SoS. The "demos" file may also be used as a canned setup file, as a basis upon which other setup files can be built. If the setup file name is correct, the SoS proceeds to load the setup file and informs the operator by displaying a "Loading setup screen" message window (Figure 24).

At any point of this Load stage, before the "Loading setup screen" message appears, the ESC key will cancel all actions and return the SoS to the Main Selection screen. A

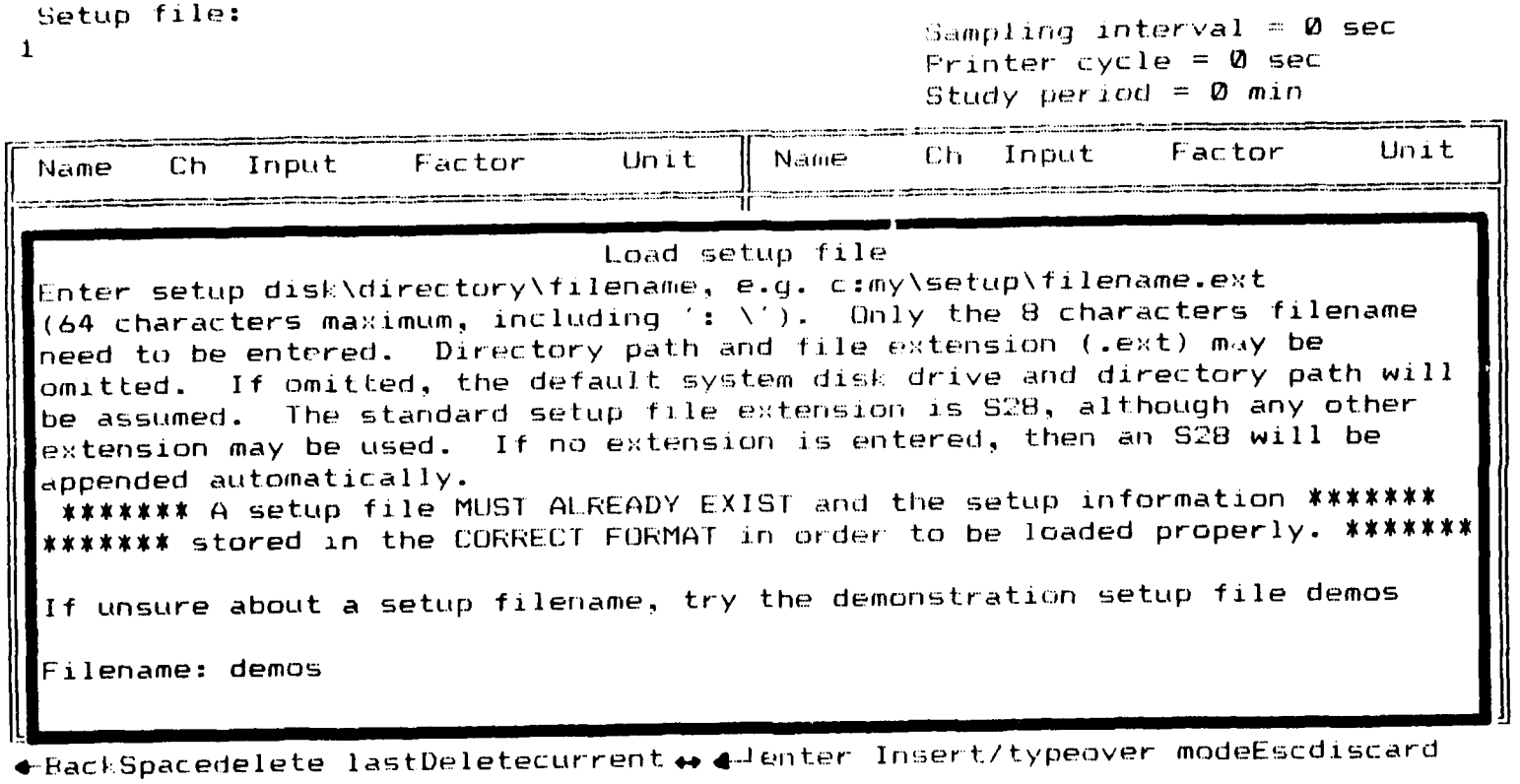


Figure 22 Load setup screen dialog box

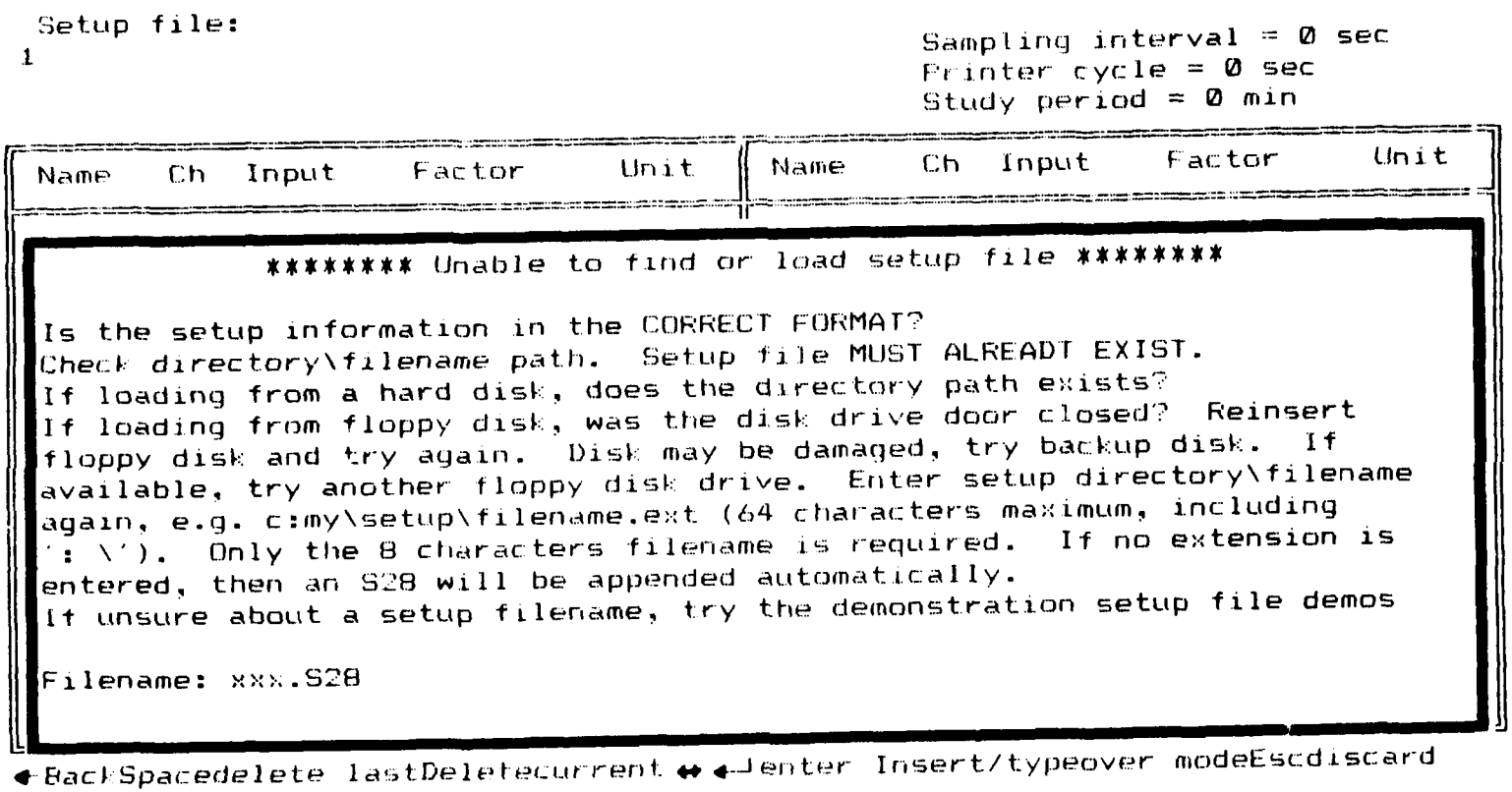


Figure 23 Load setup file error message and dialog box

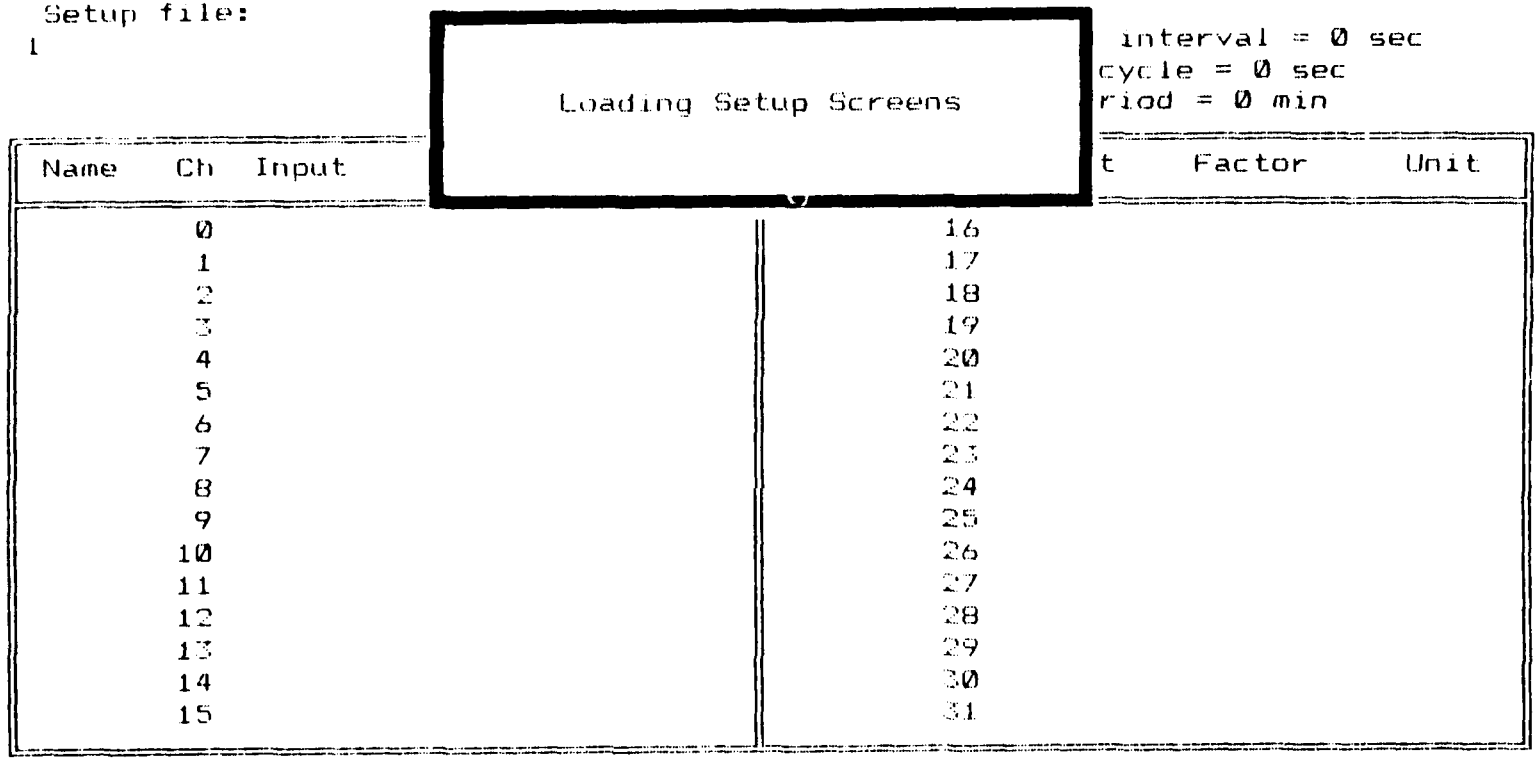


Figure 24 Loading setup screens message window

successful Load cannot be reversed, however. The Load action replaces all information in the current active setup file, thus if the current active file needs to be retained, save it first (using the F5-key) before Loading.

Save Channel Setup <SaveSetup> F5–Save setup

This option saves a permanent copy of the current active setup file. Usually, the file will be saved onto a hard disk or a floppy disk. However, the save may be to any storage medium connected as a logical unit to the PC. The operator will be prompted with extensive information about the Saving procedure (Figure 25). If an error occurs, the operator will be asked to retry, with suggestion of the possible causes of error (Figure 26). The SoS will display a "Saving setup screen" message window (Figure 27) while saving to the PC. A "setup file xxx SAVED" message in the Prompt screen (Figure 28) informs the operator of a successful save. As seen in Figure 28, the operator is also reminded to print a copy of the setup screen for hardcopy record. Recall there are four setup pages, a complete setup screen record requires printing each page separately.

Save Collected Data <SaveData> F7–Save data

Saves a permanent copy of the collected data. Data collected during data acquisition is retained temporarily on a ramdisk, which unless saved onto a more permanent medium, cannot be recovered once the PC power is turned off. Usually, the save will be to a hard disk or a floppy disk, although any logical device connected to the PC may be used.

The saving action copies a SOURCE file to a DESTINATION location. The operator will be asked to specify the SOURCE filename (Figure 29, top), and the DESTINATION filename (Figure 29, bottom). The SoS will display a "Saving Collected data" message window (Figure 30). For a successful save, the Prompt screen will display status of a successful Save, such as "1 File copied" (Figure 31, top). For an unsuccessful save, the Prompt screen will display an error message indicating the cause of error, for example "File not found" (Figure 31, bottom).

This submodule can also be called from inside the Data Acquisition module, and is more useful at that stage. At the Main Selection stage, data save can be more easily accomplished by exiting to the DOS shell and using the DOS copy command.

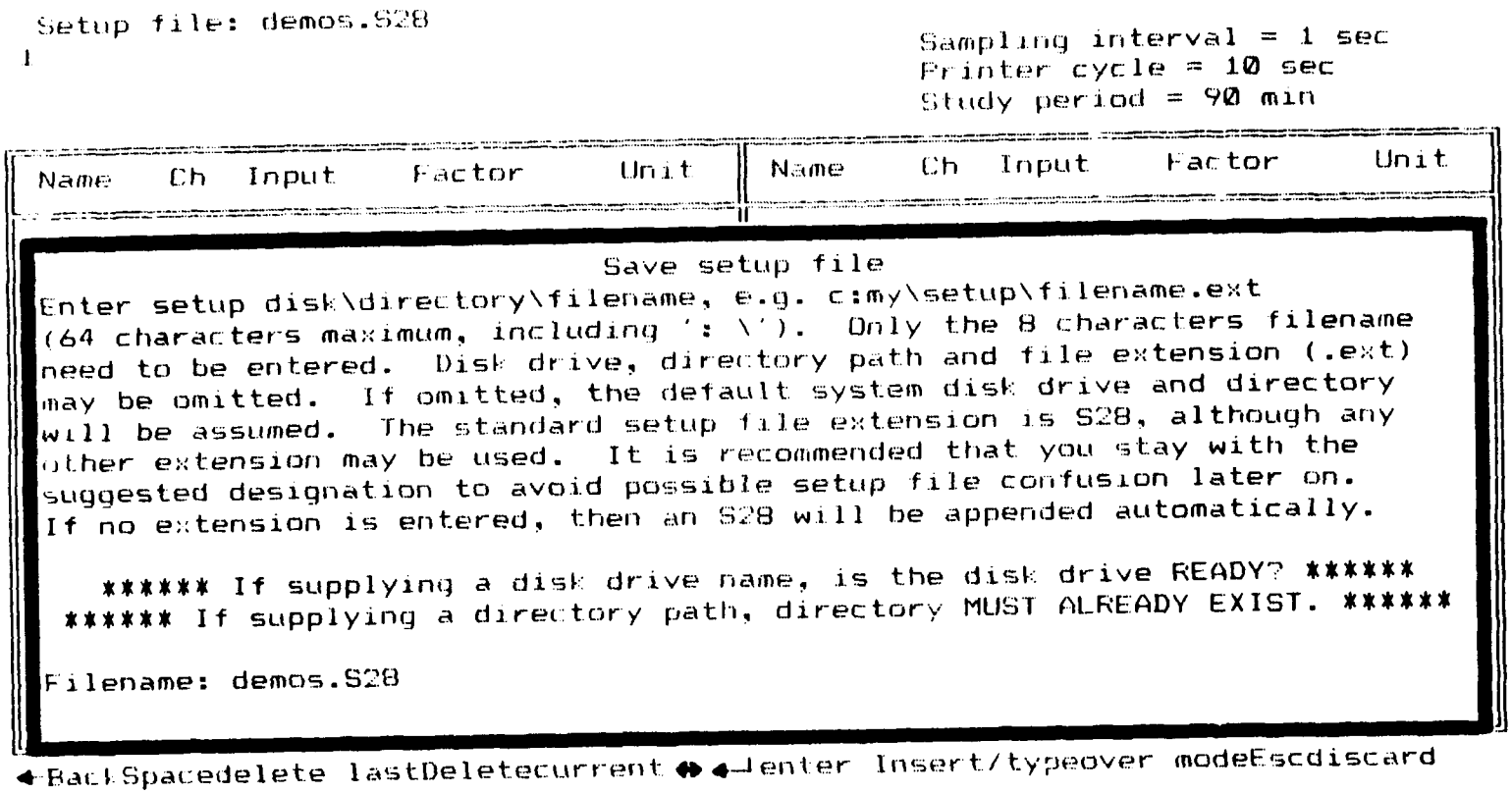


Figure 25 Save setup file dialog box

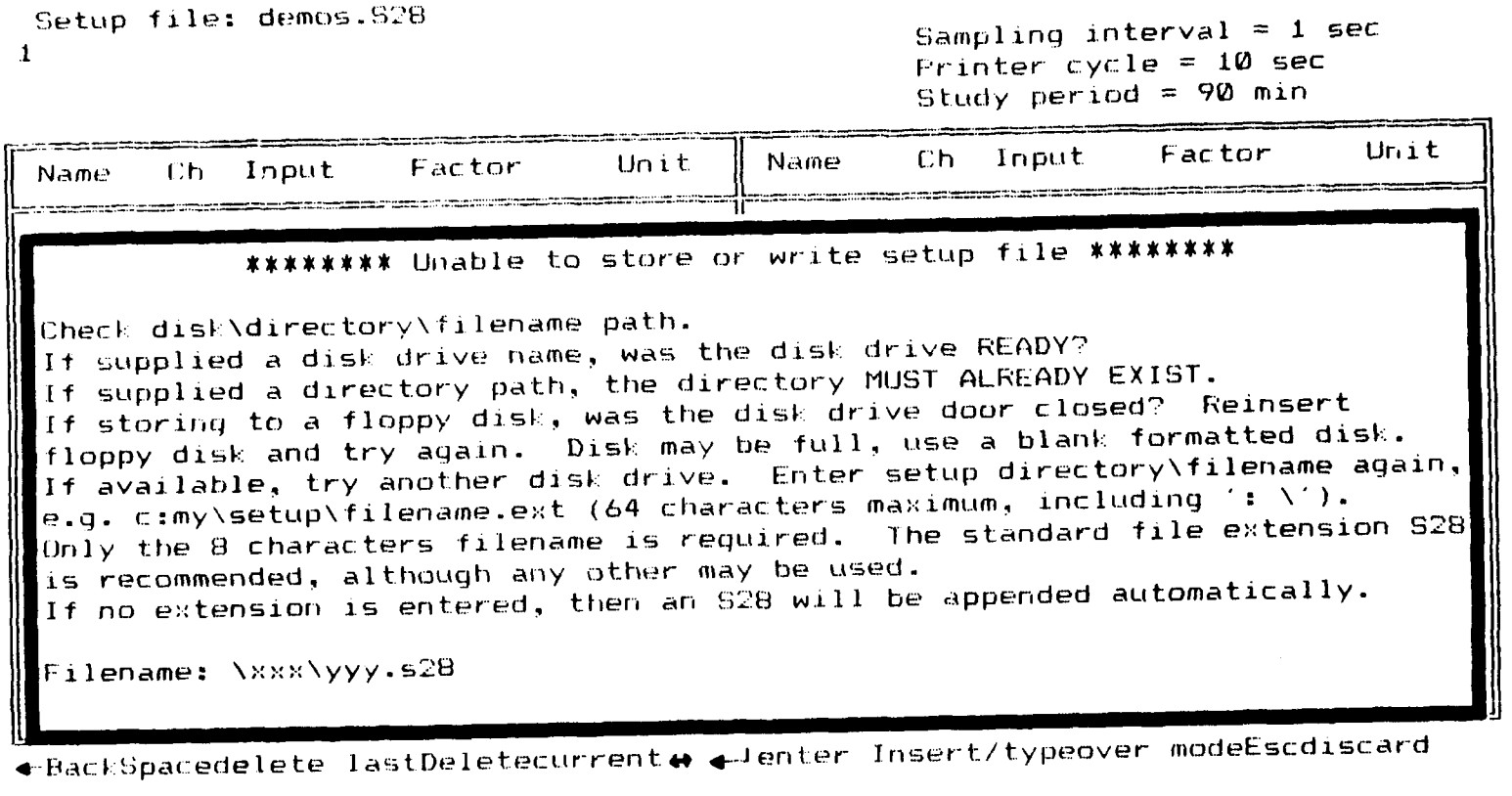


Figure 26 Save setup file error message and dialog box

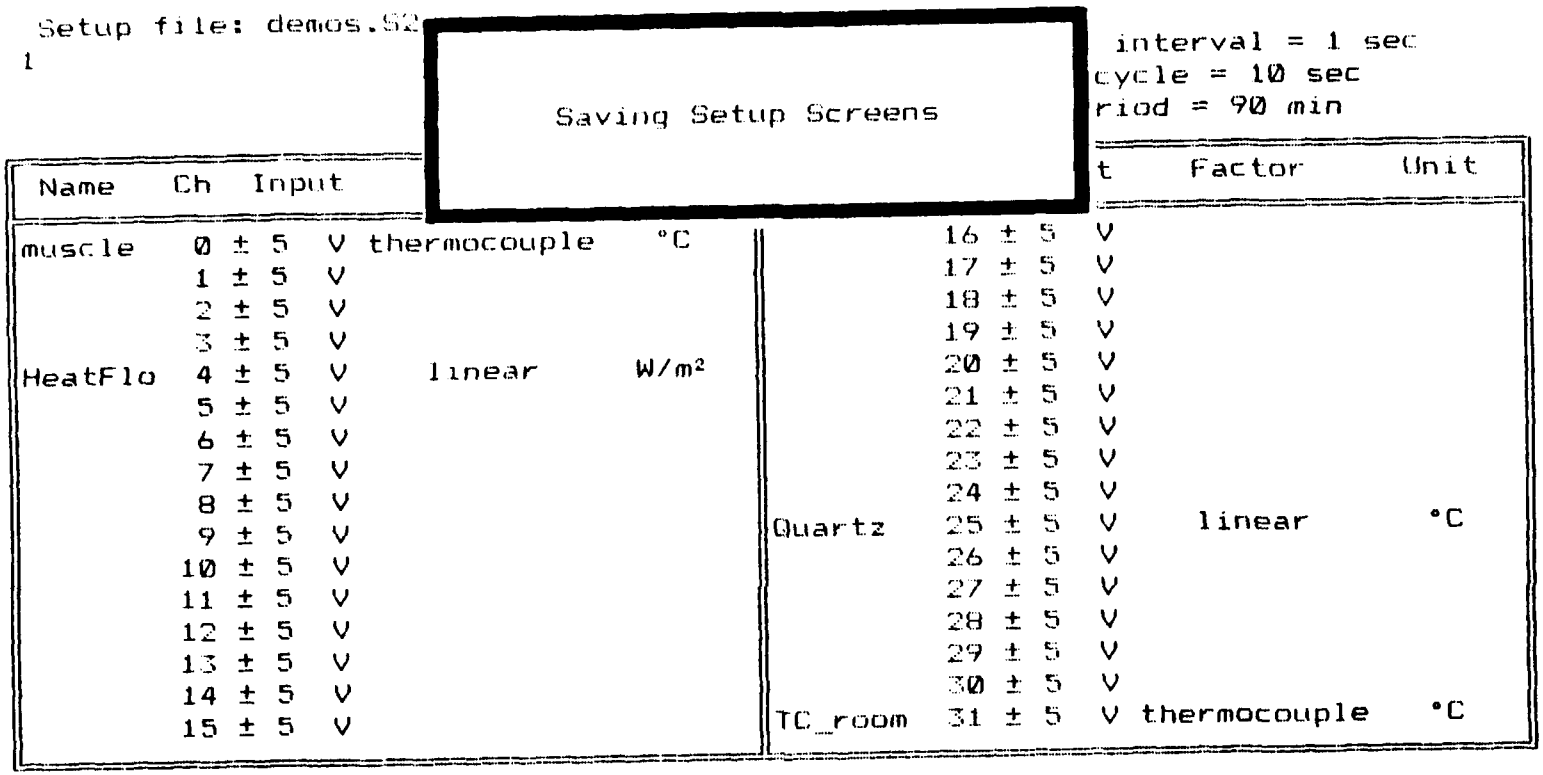


Figure 27 Saving setup screens message window

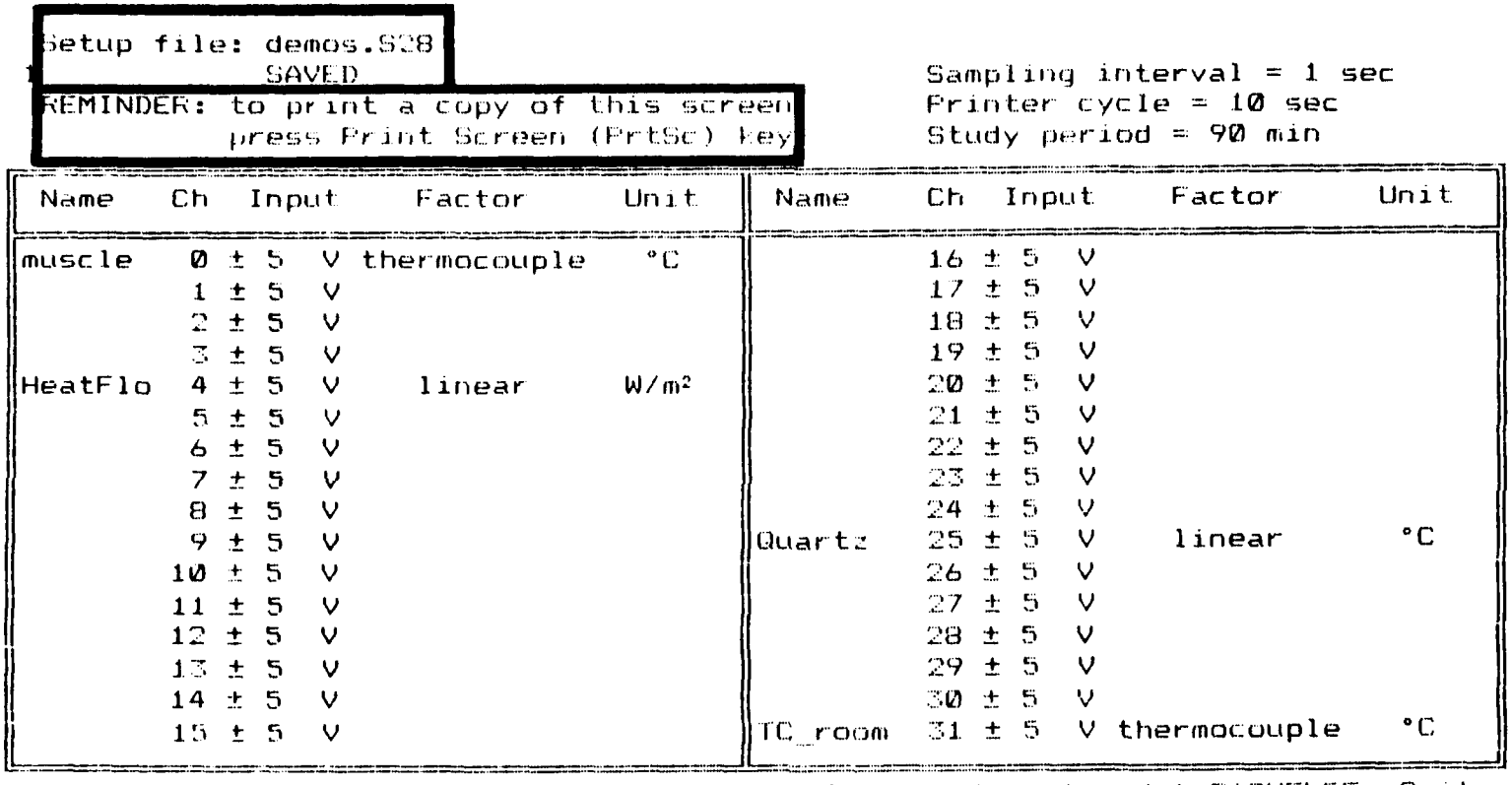


Figure 28 Setup file successfully saved message and Print Screen reminder

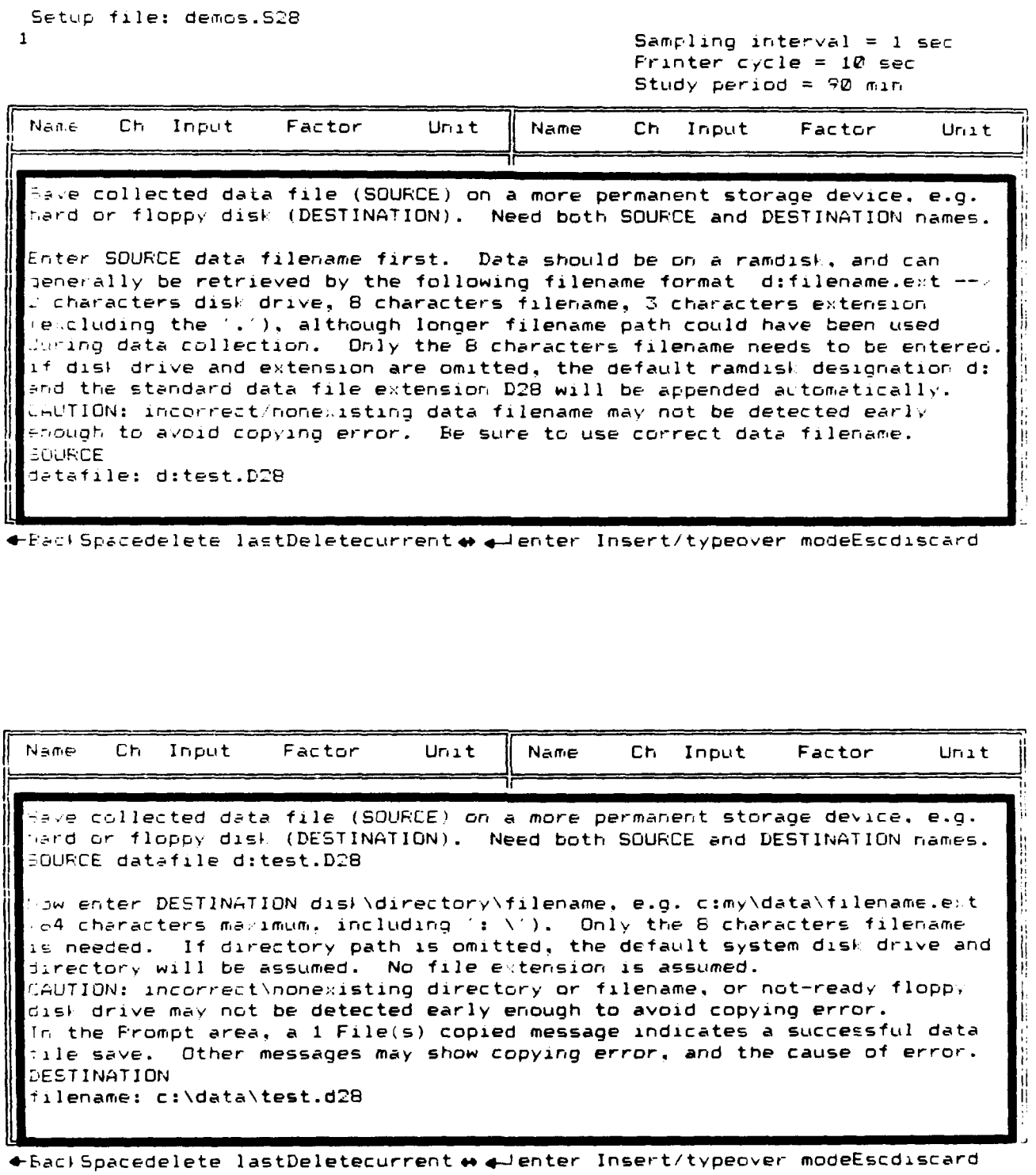


Figure 29 Save collected data SOURCE and DESTINATION filename entry dialog boxes

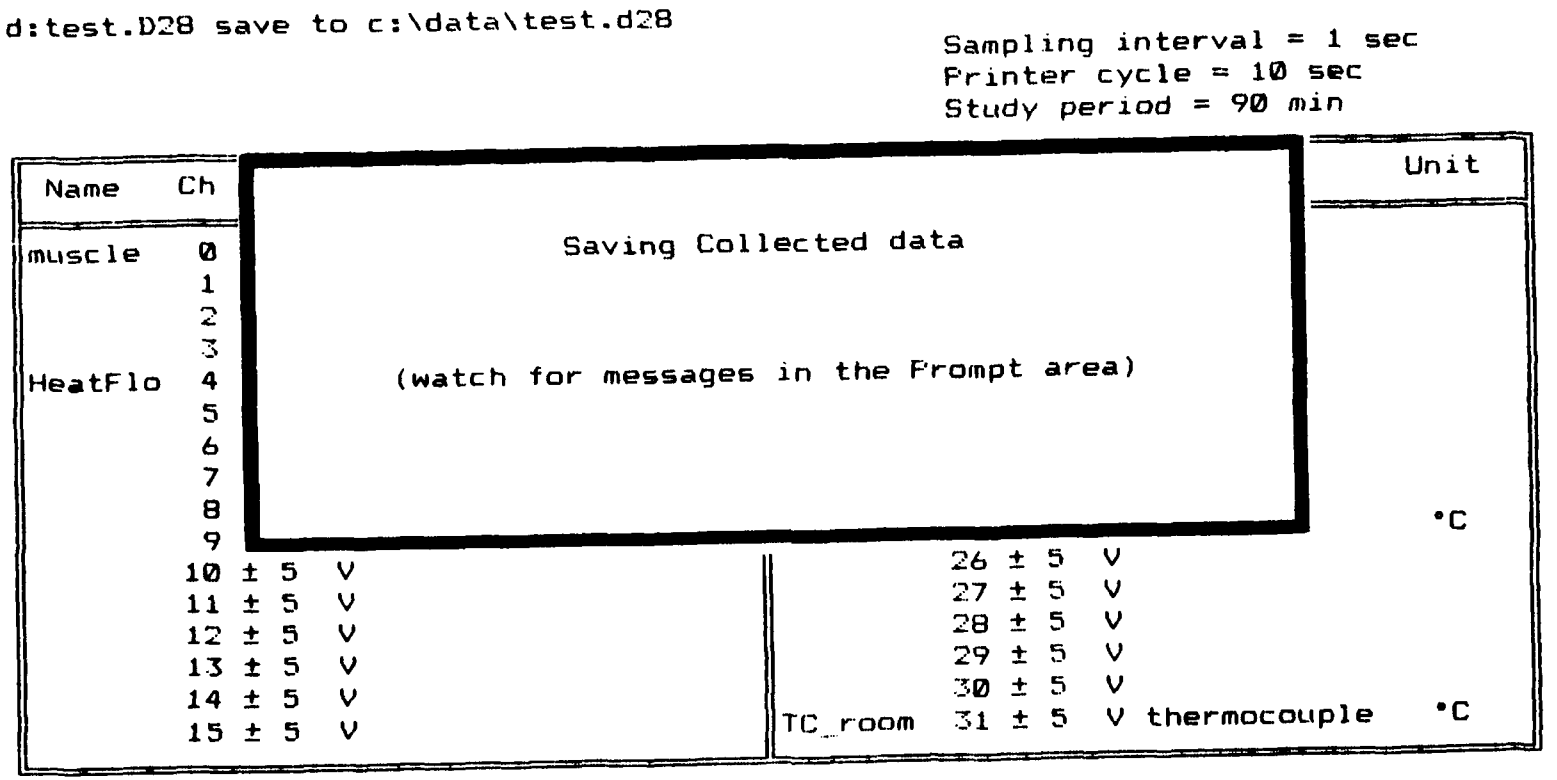


Figure 30 Saving collected data message window

```
d:test.D28 save to c:\data\test.d28
        1 File(s) copied
```

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
muscle	0	± 5 V	thermocouple	°C		16	± 5 V		
	1	± 5 V				17	± 5 V		
	2	± 5 V				18	± 5 V		
	3	± 5 V				19	± 5 V		
HeatFlo	4	± 5 V	linear	W/m²		20	± 5 V		
	5	± 5 V				21	± 5 V		
	6	± 5 V				22	± 5 V		
	7	± 5 V				23	± 5 V		
	8	± 5 V				24	± 5 V		
	9	± 5 V			Quartz	25	± 5 V	linear	°C
	10	± 5 V				26	± 5 V		
	11	± 5 V				27	± 5 V		
	12	± 5 V				28	± 5 V		
	13	± 5 V				29	± 5 V		
	14	± 5 V				30	± 5 V		
	15	± 5 V			TC_room	31	± 5 V	thermocouple	°C

F1LoadF2NewF4ModifyF5Save setupF7Save data ‡ F8DOSF9Acquire dataF10HELPEscQuit

```
d:test.D28 save to c:\data\test.d28
File not found - D:TEST.D28
        0 File(s) copied
```

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit

Figure 31 Successful and unsuccessful save data messages

DOS Shell <DosShell> F8–DOS

The AC128 SoS temporarily yields to PC DOS. Note: the PC must be able to locate a copy of the DOS COMMAND.COM file (usually on the default drive), otherwise the PC system will crash and lose all unsaved setup information and acquired data. In DOS shell, copying of data files can be more easily and quickly completed by using the DOS copy command. To return to the AC128 SoS, type "exit" (without the quotation marks) at the DOS prompt.

Main Selection HELP <MainHelp> F10–HELP

The Main Selection stage HELP is composed of two screens. Screen 1 <MainHelpPage1> contains an abbreviated version of the description of the four areas of each display page (Figure 32, top). A brief explanation of each activated F-key is available on screen 2 <MainHelpPage2> (Figure 32, bottom).

Modify Current Setup <ChannelDisplaySetup128> F4–Modify

The current active setup file may be modified. The Modify F-key throws the SoS into the Channel Setup module, which will be discussed later.

New Setup <NewSetup> F2–New

Prepare for entries into a new setup file. This submodule essentially calls the Initial Setup module. All setup information currently on screen (in all four setup pages) will be erased. Hence, if the current setup information should be retained for future use, activate F5 – Save Setup first, before proceed with <NewSetup>. The SoS is then routed to the Channel Setup module.

Acquire Data <DataAcq> F9–Acquire data

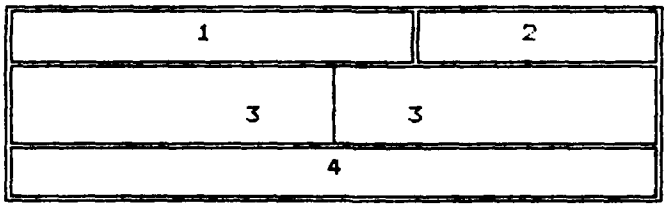
When the channel setup is completed, data acquisition can commence. A status flag, <DACQ> is set to 1 to tell the SoS to go to the Data Acquisition module.

Quit AC128 Program <ProgramExit> ESC-Quit

An ESC key at this stage exits the operator from the AC128 SoS. In the Prompt screen, the operator will be reminded to save the setup file, and the collected data (Figure 33). The operator must confirm the Quitting action. Any answer other than "y" or "Y" to the quitting confirmation question will return SoS to the Main Selection module.

GENERAL PURPOSE DATA ACQUISITION PROGRAM (128.20) - 128 channels available
Each data channel must be setup before data collection can begin. Setup procedure involves selecting the input data type and indicating the output data processing for each channel. Each channel setup is begun by choosing the appropriate F-key function, and then follow the prompt instructions.

Each screen is divided into 4 areas (not drawn to proportion)



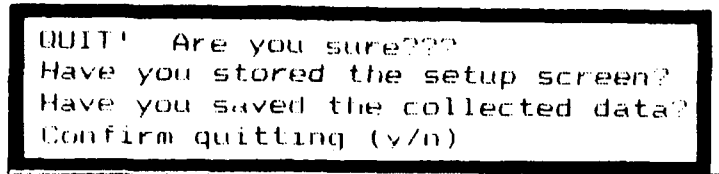
There are 4 display screens each contains 32 channels of setup information

Area (1) is the PROMPT screen. System status, acquisition channel information, program warning and error messages are displayed in this area.
Area (2) is the TIME/CYCLE screen. Data channel sampling interval, study period, and printer output cycle informations are shown.
Area (3) is the channel setup display screen. This area (each half) is divided into 5 columns: Device Name, Channel Number, Input Voltage Range, Conversion Factor, and Unit Label. The column currently active has a highlighted heading, and a blinking label indicates the currently active channel.
Area (4) is the F-key row. When this row is shown, only the indicated keys are activated. Other keys have no effect.
....MORE.... or PgDn for next screen Esc or F10 to leave Help screen

When the F-key row is shown (always the last row on the screen), only the indicated keys are activated. F-keys select particular functions. Arrows move between columns and rows. PgUp and PgDn move between the 4 setup display screens. The Esc key always exits to the previous step. Before exiting the program, you will be prompted for confirmation and reminded to save everything. Note: setup information not intentionally saved will be lost.
F1 - loads old setup file. File must already exist (i.e. saved previously). If unsure about a filename, try the demonstration setup file demos.
F2 - Use F2 to enter setup information into a new (blank) setup screen.
F4 - modifies current setup screen. When there is setup information already on the screen (loaded previously), the setup information of each data channel can be modified. New setup file can be created by modifying an old setup screen, then saving the updated screen into a new setup file.
F5 - stores the current setup screen into a setup file. You will be asked for a setup filename. Channel setup information not saved will be lost.
F7 - saves collected data into a data file. This option is also available during data acquisition. Save all the collected data as soon as possible.
F8 - goes to a DOS shell. Enter 'EXIT' to return to Data Acquisition program.
F9 - starts the data acquisition procedure. Start acquire data only when channel setup is completed. You will be prompted for a data filename.
F10 - Different HELP screens are available. Each setup column: Device Name, Channel Number, Input Voltage Range, Conversion Factor, & Unit Label has its own HELP screen. Whenever stuck, try the F10 - HELP screen.
or PgUp for previous screen Esc or F10 to leave Help screen

F1Load F2New F4Modify F5Save setup F7Save data ↕ F8DOS F9Acquire data F10HELP EscQuit

Figure 32 Main Selection stage HELP screens



Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
muscle	0	± 5 V	thermocouple	°C		16	± 5 V		
	1	± 5 V				17	± 5 V		
	2	± 5 V				18	± 5 V		
	3	± 5 V				19	± 5 V		
HeatFlo	4	± 5 V	linear	W/m²		20	± 5 V		
	5	± 5 V				21	± 5 V		
	6	± 5 V				22	± 5 V		
	7	± 5 V				23	± 5 V		
	8	± 5 V				24	± 5 V		
	9	± 5 V			Quartz	25	± 5 V	linear	°C
	10	± 5 V				26	± 5 V		
	11	± 5 V				27	± 5 V		
	12	± 5 V				28	± 5 V		
	13	± 5 V				29	± 5 V		
	14	± 5 V				30	± 5 V		
	15	± 5 V			TC room	31	± 5 V	thermocouple	°C

Figure 33 Exiting AC128 SoS confirmation message

CHANNEL SETUP <ChannelDisplaySetup128>

The ChannelDisplaySetup128 module is written as a subprogram in QuickBASIC. The subprogram is divided into five main submodules <DeviceName>, <ChannelSelect>, <InputGain>, <ConFactor>, and <UnitLabel>. These submodules corresponding to the five columns in the display screen Area (3). Figure 16 is the Channel Setup flow chart.

Arrival inside the ChannelDisplaySetup128 subprogram is signified by one of the setup entries on the display screen starting to blink. The blinking entry indicates the current active entry. The initial default active entry will be channel 0 in the Channel Select (Ch) column, i.e. the number "0" will start to blink. Any action or task selected using the F-key always operates on the current active entry. ↑ and ↓ arrows move the active (blinking) entry to another channel, but stay in the same column field. → and ← move the active entry to another column, and the active channel number stays the same. PageUp (PgUp) and PageDown (PgDn) bring up another page (another 32 channels) of display. The plotting screens are not available for viewing at this Channel Setup stage.

The channel setup information that appears on the same row as a channel number applies to that data channel. For example, in Figure 34, the device name (muscle), input voltage (±5V), conversion factor (thermocouple), and unit label (°C) that appear on the same row as channel 0, in the left half display page 1, apply to or are parameters of data input channel 0.

Setup Completed <SetupDone> ESC-Done

When the channel setup is completed, the SetupDone submodule reminds the operator to save the setup information and returns SoS to the Main Selection module. In Figure 35, the reminder message is shown in the Prompt screen. SetupDone is the only exit point from the Channel Setup module, and SoS then returns to the Main Selection stage. The SetupDone submodule can be called from anywhere within the Channel Setup module, i.e. from any one of the five main channel setup submodules

Setup file: demos.S28

1

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
muscle	0	± 5 V	thermocouple	°C		16	± 5 V		
	1	± 5 V				17	± 5 V		
	2	± 5 V				18	± 5 V		
	3	± 5 V				19	± 5 V		
HeatFlo	4	± 5 V	linear	W/m²		20	± 5 V		
	5	± 5 V				21	± 5 V		
	6	± 5 V				22	± 5 V		
	7	± 5 V				23	± 5 V		
	8	± 5 V				24	± 5 V		
	9	± 5 V			Quartz	25	± 5 V	linear	°C
	10	± 5 V				26	± 5 V		
	11	± 5 V				27	± 5 V		
	12	± 5 V				28	± 5 V		
	13	± 5 V				29	± 5 V		
	14	± 5 V				30	± 5 V		
	15	± 5 V			TC_room	31	± 5 V	thermocouple	°C

F1LoadF2NewF4ModifyF5Save setupF7Save data ↕ F8DOSF9Acquire dataF10HELPEscQuit

Figure 34 Current active data channel setup parameters

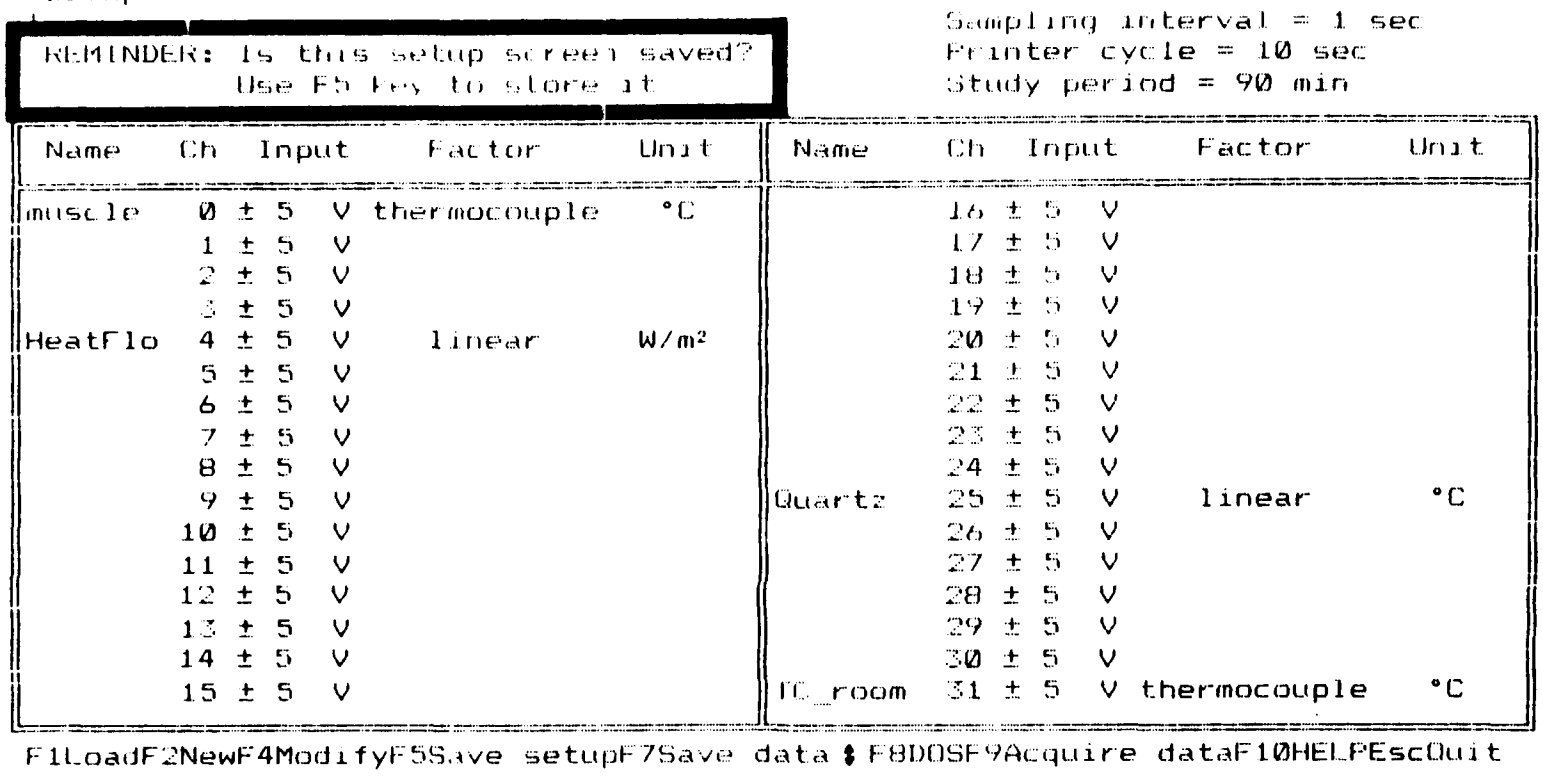


Figure 35 Setup done and save setup file reminder

Device Name Submodule <DeviceName>

Figure 36 is an example of the Device Name screen. "Device Name Selection" appears in the Prompt screen. The "Name" column heading is highlighted, and the current active device name starts to blink. The F-key row shows the available function key options.

Enter Device Name F1–Enter Name. Enter a name for the device connected to this channel (Figure 37, top). If a device name already exists, then it is shown in the entry dialog box. The device name is a alphanumeric name provided for the designation/identification of a channel. The name can be a maximum of 7 characters long. If a disallowed character is entered, the SoS will beep and issue an error message in the dialog box indicating which is the offending character. The illegal character is ignored and the operator is prompted to continue with the device name entry. The default device name is "none". If less then 7 characters were entered for a device name, the SoS will pad the name to be 7 characters long with the space (ASCII 32) character. This is done to align the printer output.

The operator may also replace the current active entry name with the device name from another channel (using the F5-replace key). The SoS will open a replacement window inside the Enter Name dialog box (Figure 37, bottom) to ask for the replacement channel number. Afterwards, both channels will have the exact same device name. This option is useful if the same device name is used for more than one data channel, or if the device names are in a serial form

Remove Device Name F8–Remove Name. Remove the device name designation from this data channel. The device name will be replaced with "none".

Setup Completed or Exit F9/Esc–Done/Exit. This calls the SetupDone submodule to exit the Channel Select module and returns SoS to the Main Selection stage.

Device Name HELP Screen <DeviceNameHelp> F10–HELP. See Figure 38. A portion of this HELP screen <NameUnitHELP1> is shared with the Unit Label HELP screen.

Setup file: demos.S28

1

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

Device Name Selection

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
muscle	0	± 5 V	thermocouple	°C		16	± 5 V		
	1	± 5 V				17	± 5 V		
	2	± 5 V				18	± 5 V		
	3	± 5 V				19	± 5 V		
HeatFlo	4	± 5 V	linear	W/m²		20	± 5 V		
	5	± 5 V				21	± 5 V		
	6	± 5 V				22	± 5 V		
	7	± 5 V				23	± 5 V		
	8	± 5 V				24	± 5 V		
	9	± 5 V			Quartz	25	± 5 V	linear	°C
	10	± 5 V				26	± 5 V		
	11	± 5 V				27	± 5 V		
	12	± 5 V				28	± 5 V		
	13	± 5 V				29	± 5 V		
	14	± 5 V				30	± 5 V		
	15	± 5 V			TC_room	31	± 5 V	thermocouple	°C

F1Enter Name F8Remove Name PgUp ← ↑ ↓ → PgDn F9Done F10HELP EscExit

Figure 36 Device Name submodule screen

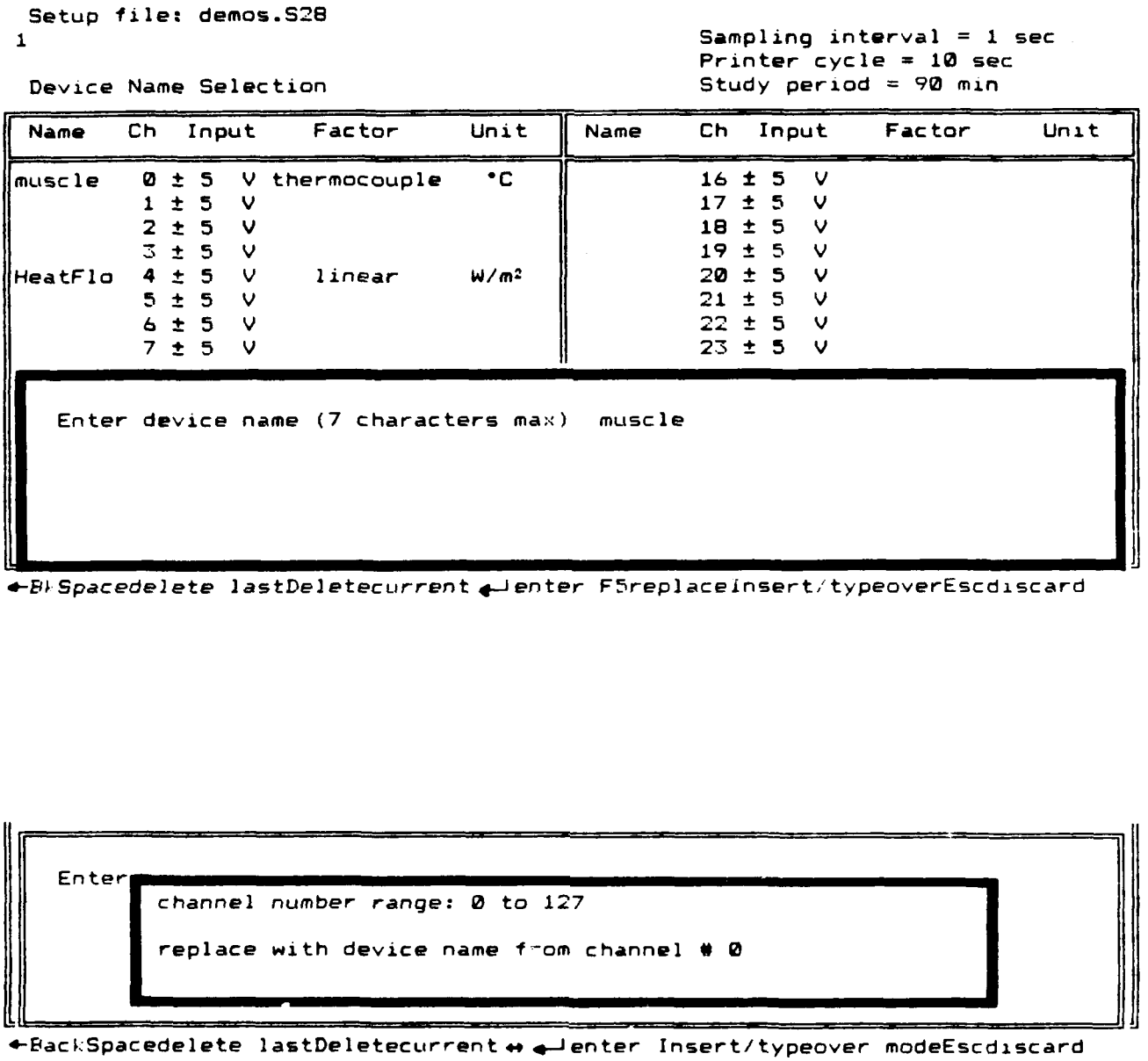


Figure 37 Enter device name dialog box and device name replacement window

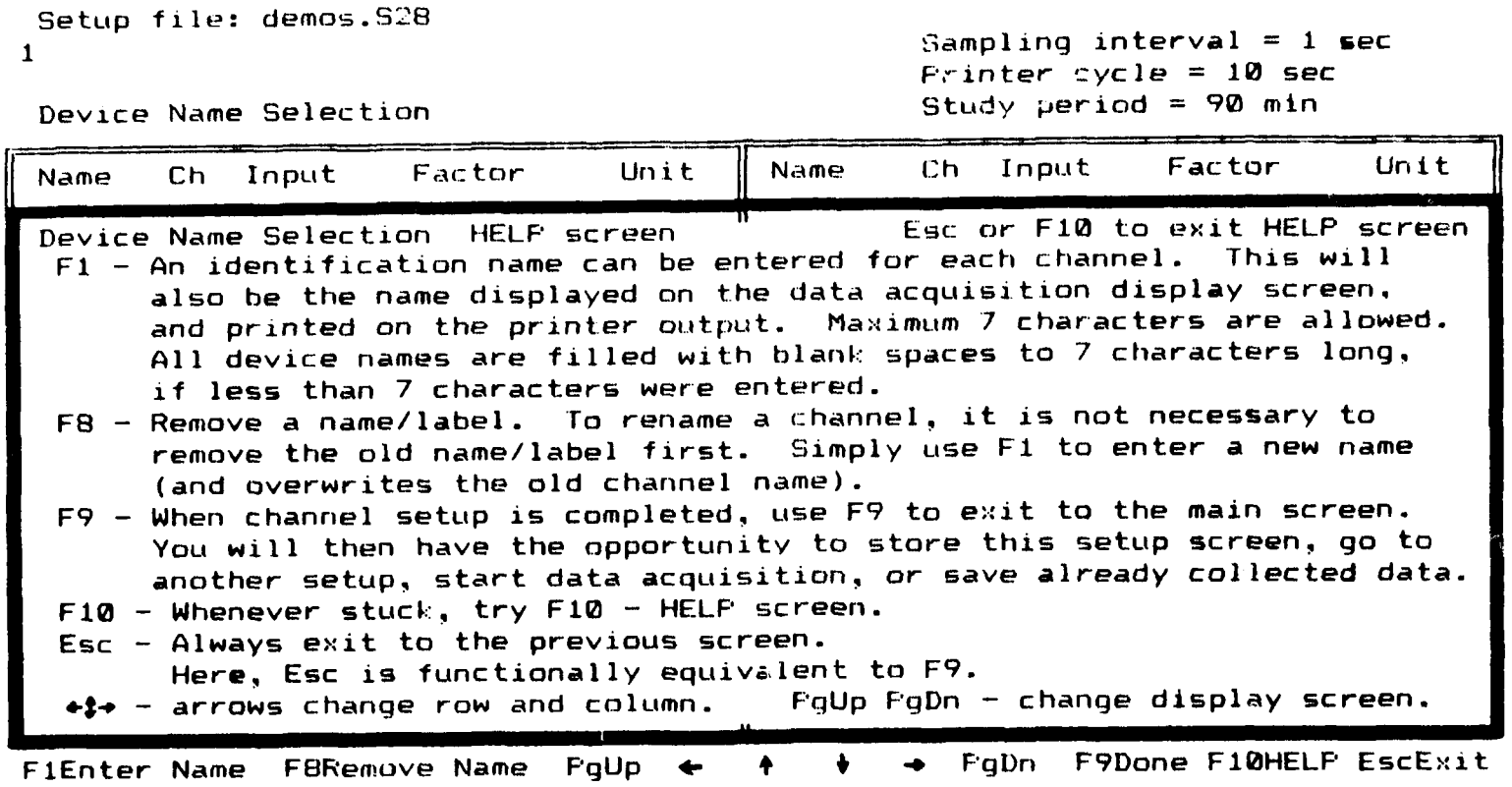


Figure 38 Device Name submodule HELP screen

Channel Select Submodule <ChannelSelect>

A more detailed flow chart of the submodule is shown is Figure 17. Figure 39 is an example of the Channel Select screen. "Channel Parameter Selection" appears in the Prompt screen. The "Ch" column heading is highlighted and the current channel number starts to blink. The F-key row shows the available options.

Enable Data Channel F5–Enable. A channel must be enabled in order to collect data. A brightened channel number on the display screen indicates an enabled channel. If a channel has already been enabled or selected for data plotting previously, then this option will have no effect. It's a good practice to enable only the channels with valid input data. If a data channel is not needed or has no input connection, disable it. This ensures a more efficient data acquisition routine.

Disable Data Channel F8–Disable. A disabled channel is indicated by a darkened channel number on the display screen. Disabling a channel, however, does not automatically remove other channel parameters e.g. device name, input voltage, conversion factor, or unit label. A disabled channel simply will not collect data. This action also stops data plotting if plotting was previously selected.

Setup Completed or Exit F9/Esc–Done/Exit. This calls the SetupDone submodule to exit the Channel Select module and returns SoS to the Main Selection stage.

Channel Select HELP Screens <ChannelSelectHelp> F10–HELP. Two HELP screens are available. Screen 1 <ChannelSelectHelpPage1> explains the data acquisition timing parameters (Figure 40, top). Screen 2 <ChannelSelectHelpPage2> explains the other F-key functions (Figure 40, bottom).

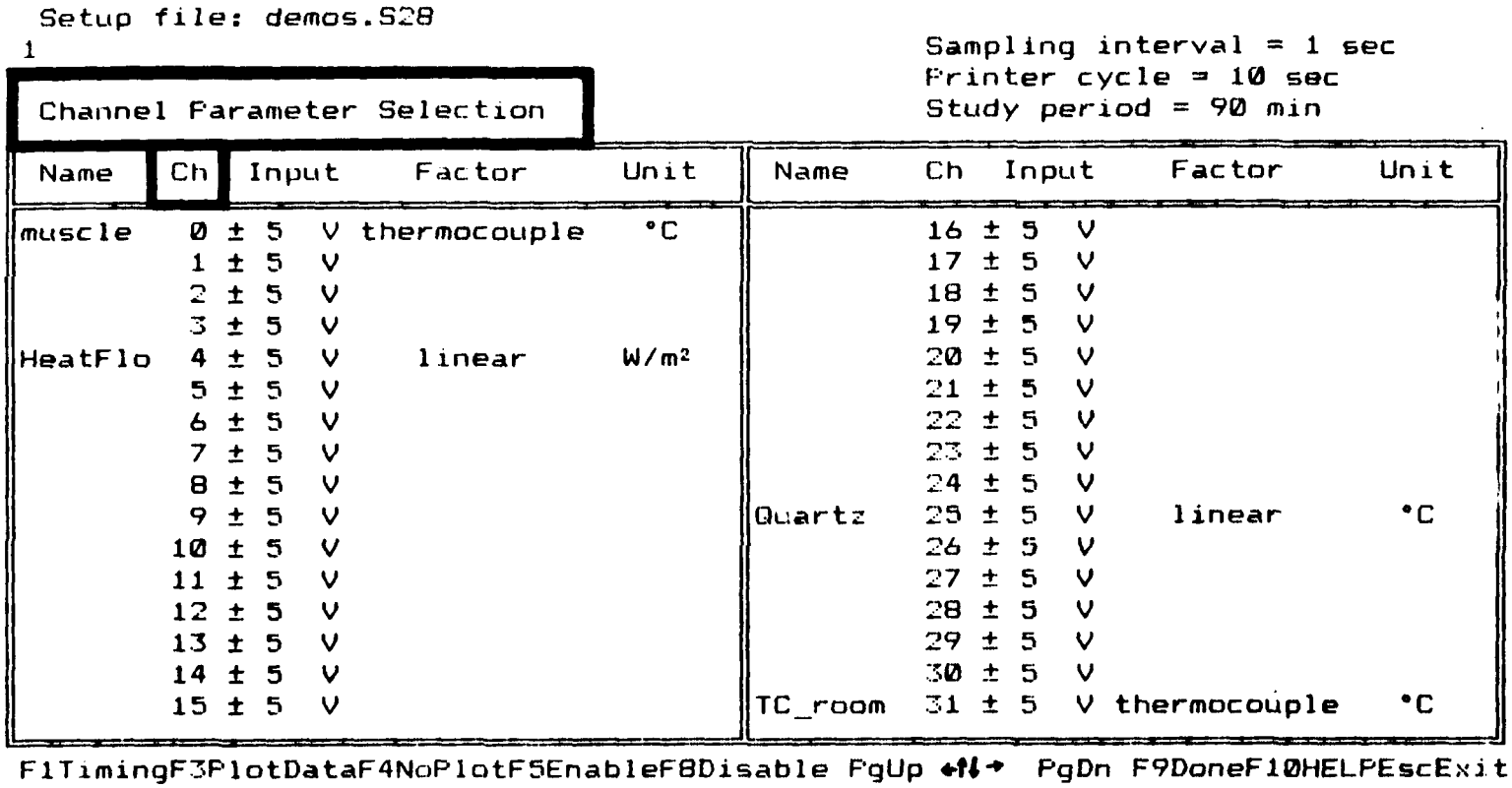


Figure 39 Channel Select submodule screen

```
 Setup file: demos.S28
1                                                  Sampling interval = 1 sec
                                                   Printer cycle = 10 sec
 Channel Parameter Selection                       Study period = 90 min

| Name   Ch  Input    Factor      Unit | Name   Ch  Input    Factor      Unit |

Channel Parameters  HELP screen                    Esc or F10 to exit HELP screen
 F1 - data acqusition Timing information.  Selection is provided for <1>
      data sampling interval, <2> printer output cycle, and <3> study period
     <1> Sampling interval - gives the frequency (in seconds) of each round
           of data collection.  Data (e.g. temperatures, voltages) from every
           ENABLED channels are collected once every sampling interval.  All
           of the collected data are stored in the computer, and the option
           for saving collected data permanently onto a disk is provided.
     <2> Printer cycle - indicates how often (in seconds) the sampled data
           are to be output to the printer, providing a hardcopy record.  For
           each channel, the printer data represents an average of all data
           sampled since the last printout.  This average data is also
           displayed under the Data Acquisition screen column 'Average'.
     <3> Study period - is the length of the study session in minutes.
 ←↕→ - arrows change row and column.     PgUp PgDn - change display screen.
 ....MORE....   ↓ or PgDn for next screen     Esc or F10 to leave Help screen

F1TimingF3PlotDataF4NoPlotF5EnableF8Disable PgUp←↑↓→ PgDn F9DoneF10HELPEscExit
```

```
| Name   Ch  Input    Factor      Unit | Name   Ch  Input    Factor      Unit |

 F3 - Plot data.  A time-course graphic plot of collected data which proceeds
      simultaneously with normal data display.  4 channels can be assigned to
      be displayed on 4 display screens (one on each screen).  Data plot
      selection, represented by 111, also automatically enables a channel.
 F4 - Disable data plotting. This stops data plotting but leaves ch. enabled.
 F5 - Enable a channel.  A brightened channel number 111 indicates the
      channel was enabled.  Only an enabled channel can collect data.  Dis-
      able all unused channels for a more efficient data acquisition routine.
 F8 - Disable a channel.  A darkened channel number 111 indicates a disabled
      channel.  This action disables both data collection and data plotting.
 F9 - When channel setup is completed, use F9 to exit to the main screen.
      You will then have the opportunity to store this setup screen, go to
      another setup, start data acquisition, or save already collected data.
 F10 - Whenever stuck, try F10 -  HELP screen.
 Esc - Always exit to the previous screen.  Here, Esc is equivalent to F9.
          ↑ or PgUp for previous screen     Esc or F10 to leave Help screen
```

Figure 40 Channel Select submodule HELP screens

Data Acquisition Timing Parameters <TimeCycle> F1-Timing. The data acquisition control timing information is entered. The three timing parameters are sampling interval, printer output cycle, and study period. The SoS will ask for them in this sequence. Figure 41 shows the Timing dialog boxes asking for the three parameters. These parameters must be entered by the operator in order for the data acquisition phase to function.

Sampling interval <TimeCycleSamp> gives the frequency, in seconds, of each round of data collection. The SoS will collect data from all the enabled input channels once every sampling interval. The sampling interval may be from 1 second to 24 hours.

Printer output cycle <TimeCycleSout> indicates how often, in seconds, the SoS should send the sampled data to the printer for a hard copy record. Recall that the data collected during data acquisition is temporarily stored on a ramdisk, hence, if there is for any reason a power interruption to the PC before a deliberate Save Data action, all data on the ramdisk will be lost. The printer output will then represent the only available record. For every enabled channel, the data printed will be an average of all the data collected since the last printer output. The averaging algorithm is a function of the printer cycle as well as the sampling interval. For example, if the sampling interval is every 1 second and the printer cycle is 10 seconds, then the printed data will be the average of the last 10 data samples collected since the last printer output. This averaged data is also displayed on the PC monitor under the 'Average' column. The shortest printer output cycle allowed is 10 seconds. The longest can be 24 hours.

Study period <TimeCycleSper> is the length of the study session, in minutes. The minimum study period is 1 minute. The maximum allowed study period depends on the total amount of data that can be stored on the ramdisk. Equivalently, the maximum allowed study period is a function of the number of data channel enabled and the sampling interval. The minimum required ramdisk size is two megabytes. The SoS will compute the maximum study period allowed based on the number of enabled channels and inform the operator accordingly.

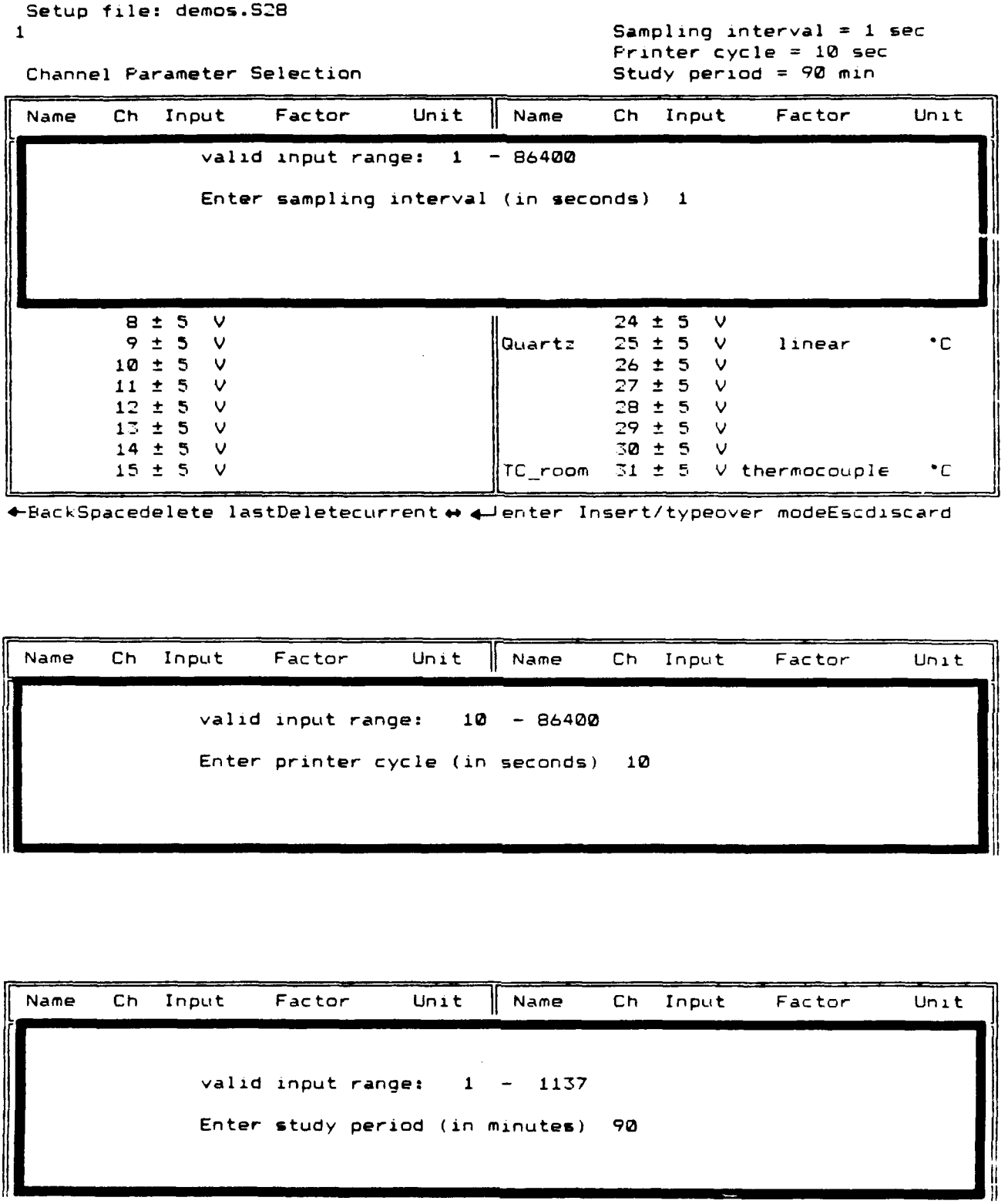


Figure 41 Data acquisition Timing parameter dialog boxes

Activate Data Plotting <PlotData> F3–Plot. A time-course graphic representation of the data acquired on the designated channel will be displayed. Figure 42 shows an example. There are four plot screens (display pages 5 through 8) available. Each can be assigned to display data collected from one input channel. The selection of data plotting will automatically enabled a channel if it was not already enabled. A channel selected for data plotting will show a bright yellow channel number.

The data plots have time as the mantissa (x-axis) and the input data units on the ordinate (y-axis). The time axis represents the entire length of the study period. The SoS will prompt the operator for the upper and lower limits of the input data (Figure 43). These upper and lower ranges will determined the scale used for the ordinate axis. In general, the smallest yet still practical range should be chosen between the upper and lower limits, to maximize the plot resolution. The unit on the ordinate axis will be the Unit Label entered for this channel. The time axis resolution, and the ordinate axis unit and resolution are displayed on the left-hand side of the plotting screens (Figure 44).

Note, this data plotting capability should be considered to be rudimentary. The resolution (determined by the upper and lower limits entered) of the display may be quite limited. It gives a good time course display of the data trend, but it should not be relied upon for quantitative data discrimination. For example, Figure 42 shows that the data plot give a reasonable indication that the measured temperature has fallen during the past hour, but it should not be used to determined that the given temperature has changed by 0.1°C in the last minute.

Deactivate Data Plotting F4–NoPlot. Data will not be plotted for this channel. The channel will remain enabled, i.e. the bright yellow channel number will change to a brightened white color. If data plotting was not previously selected for this channel then this option will have no effect.

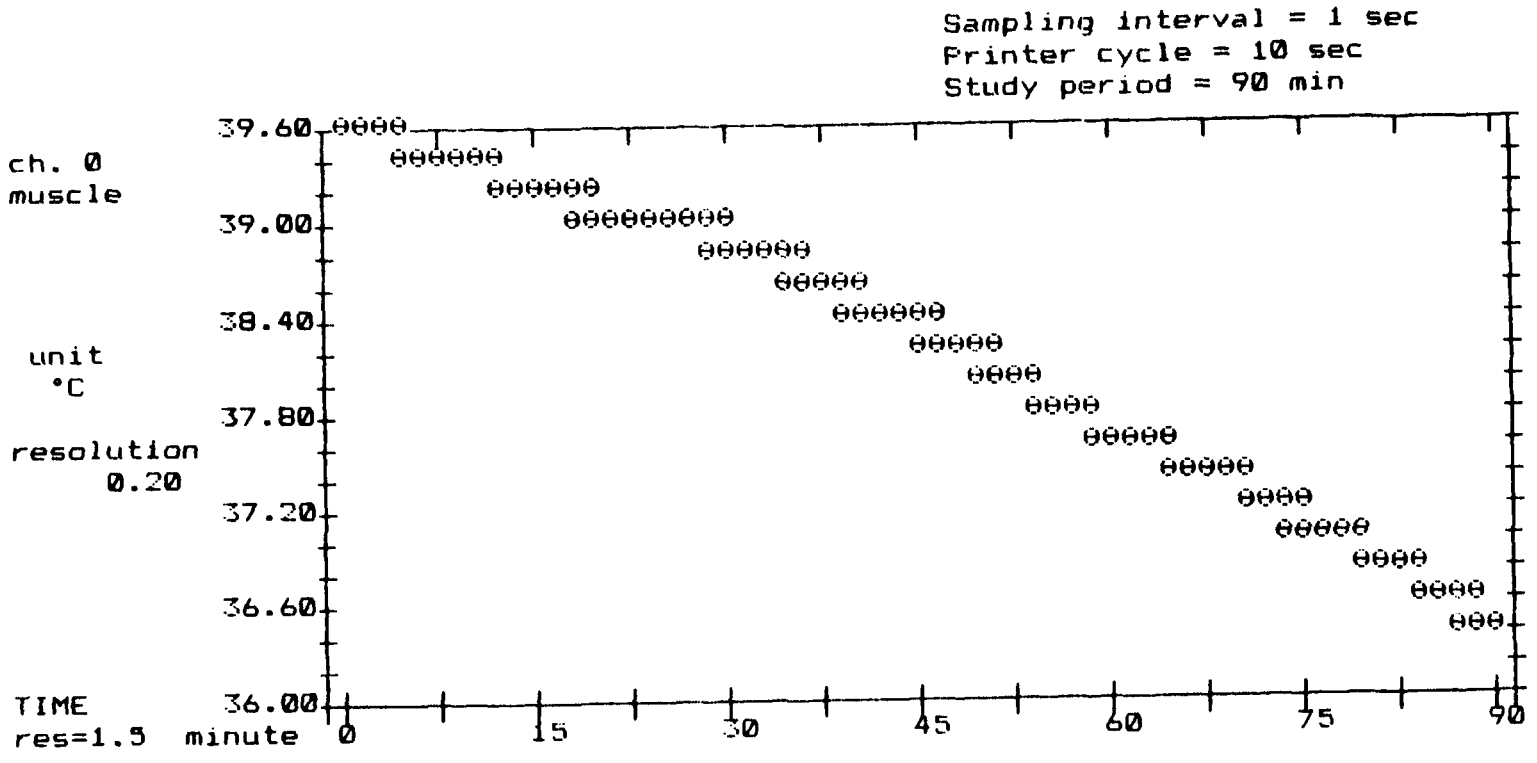


Figure 42 Sample plotting screen

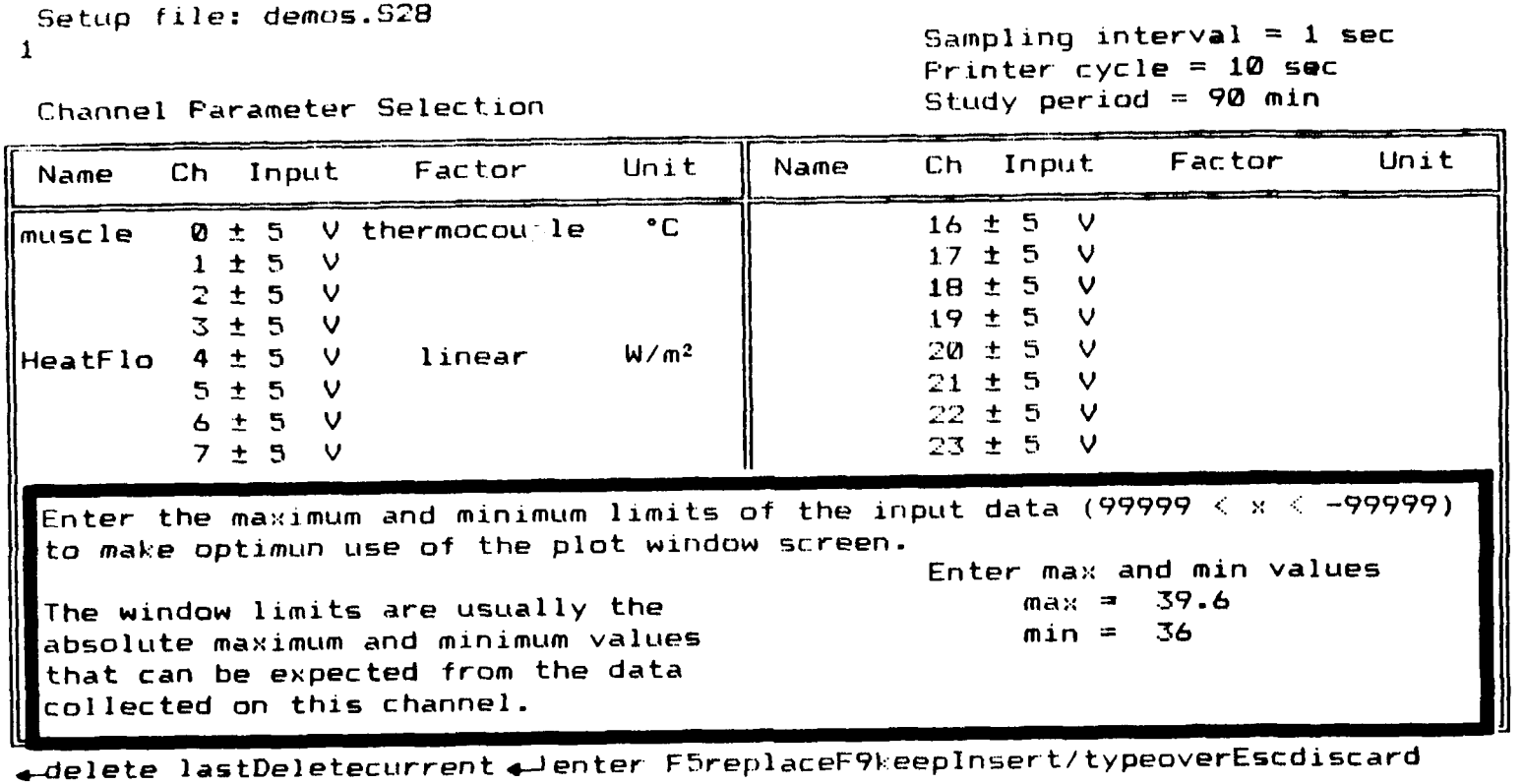


Figure 43 Plotting screen data upper and lower limits entry dialog box

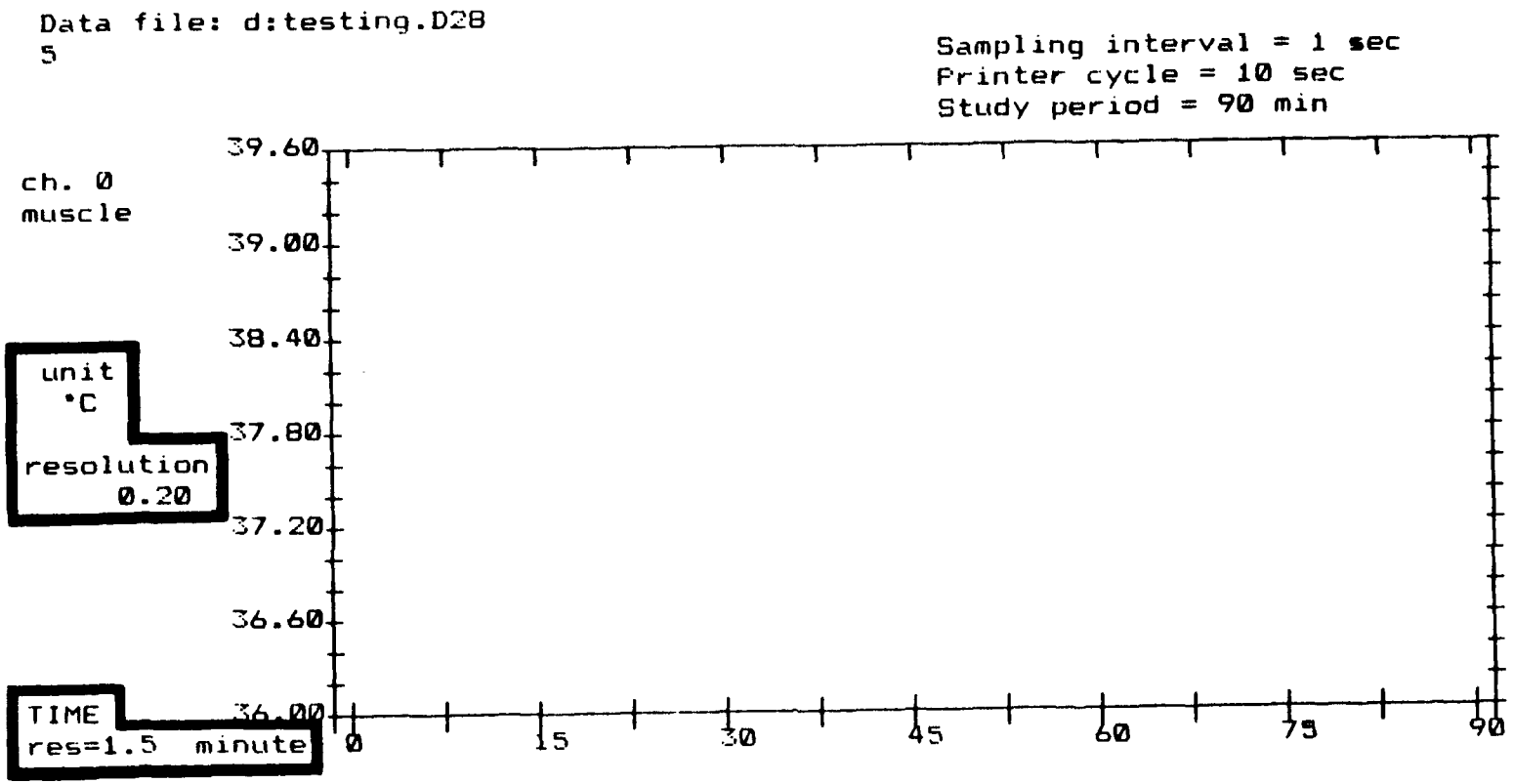


Figure 44 Plotting screen showing ordinate axis (y-axis) unit and resolution, and time axis (x-axis) resolution

Input Voltage and Gain Select Submodule <InputGain>

Figure 45 is an example of the Input Voltage screen. "Input Voltage (Gain Code) Selection" appears in the Prompt screen. When input voltage is selected by the operator, a corresponding gain code is registered in the SoS internally. The A/D board uses the gain codes to determine the input voltage range. The "Input" column heading is highlighted, and the current input voltage range starts to blink. The F-key row shows the available options.

± 5 V Input Voltage Range F1–±5V. Resolution is 2.44 mV. Resolution represents the smallest voltage difference that the hardware can accurately and reliably differentiate.

± 500 mV Input Voltage Range F2–±500mV. Resolution is 0.244 mV.

± 50 mV Input Voltage Range F3–±50mV. Resolution is 24.4 μV.

± 10 mV Input Voltage Range F4–±10mV. Resolution is 2.44 μV.

Remove Input Voltage Selection F8–Remove. Voltage select and gain code is removed. The screen will display only "±". However, if a channel is enabled or has been selected for data plotting, then the input voltage range can not be removed without disabling the channel first. The SoS will beep and issue a warning message in the Prompt screen (Figure 46) to warn the operator of such situation, and the Remove action is ignored.

Setup Completed or Exit ESC/F9–Done/Exit. This calls the SetupDone submodule to exit the C annel Select module and returns SoS to the Main Selection stage.

Input Voltage HELP Screen <InputGainHelp> F10–HELP. See Figure 47.

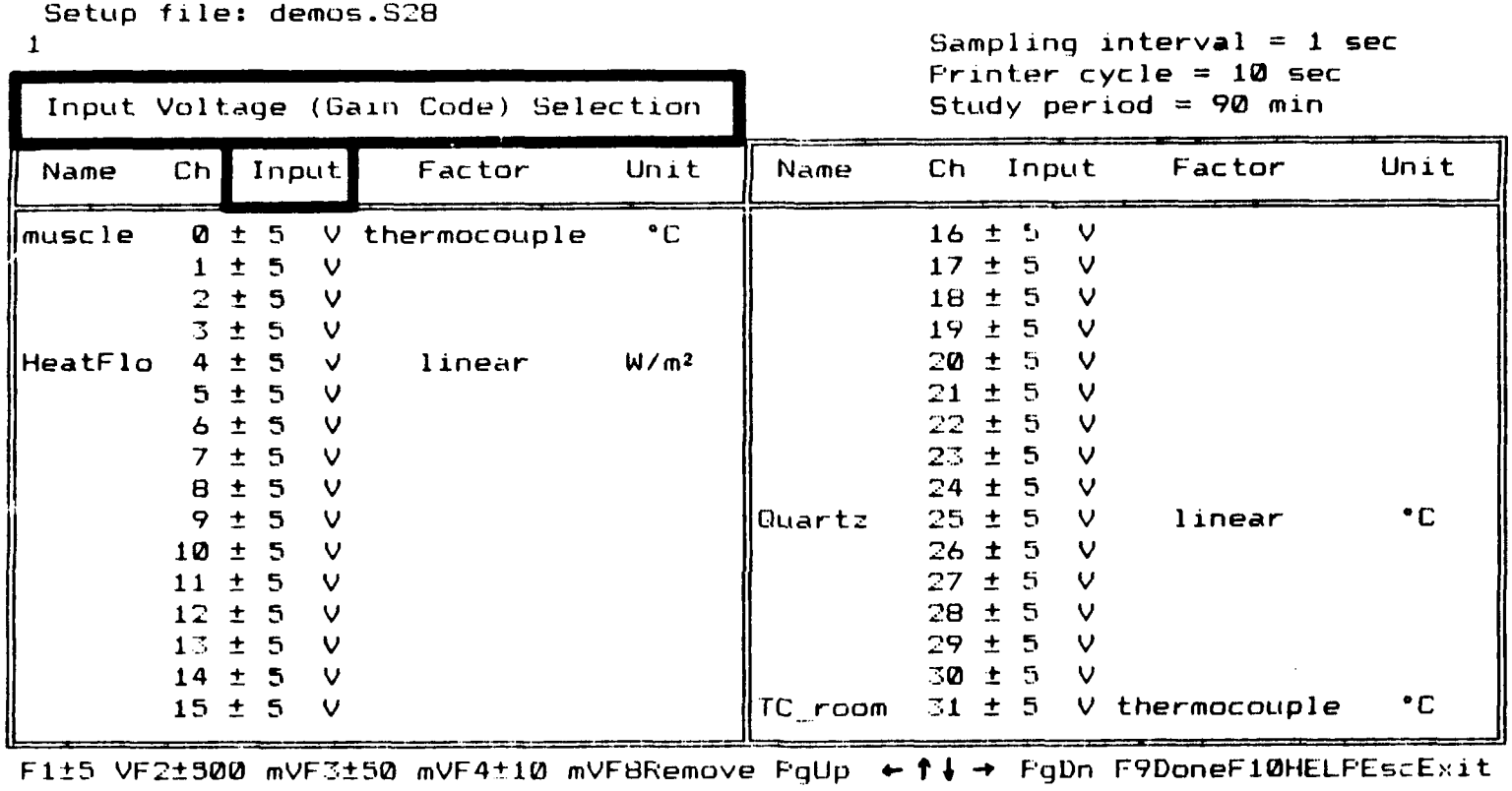


Figure 45 Input Voltage Range and Gain Code Select submodule screen

Setup file: demos.S28

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

««Channel enabled, cannot remove input voltage»»

Input Voltage (Gain Code) Selection

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
muscle	0	± 5 V	thermocouple	°C		16	± 5 V		
	1	± 5 V				17	± 5 V		
	2	± 5 V				18	± 5 V		
	3	± 5 V				19	± 5 V		
HeatFlo	4	± 5 V	linear	W/m²		20	± 5 V		
	5	± 5 V				21	± 5 V		
	6	± 5 V				22	± 5 V		
	7	± 5 V				23	± 5 V		
	8	± 5 V				24	± 5 V		
	9	± 5 V			Quartz	25	± 5 V	linear	°C
	10	± 5 V				26	± 5 V		
	11	± 5 V				27	± 5 V		
	12	± 5 V				28	± 5 V		
	13	± 5 V				29	± 5 V		
	14	± 5 V				30	± 5 V		
	15	± 5 V			TC_room	31	± 5 V	thermocouple	°C

F1±5 VF2±500 mVF3±50 mVF4±10 mVF8Remove PgUp ← ↑ ↓ → PgDn F9DoneF10HELPEscExit

Figure 46 Warning message disallowing removal of input voltage range when channel is enabled

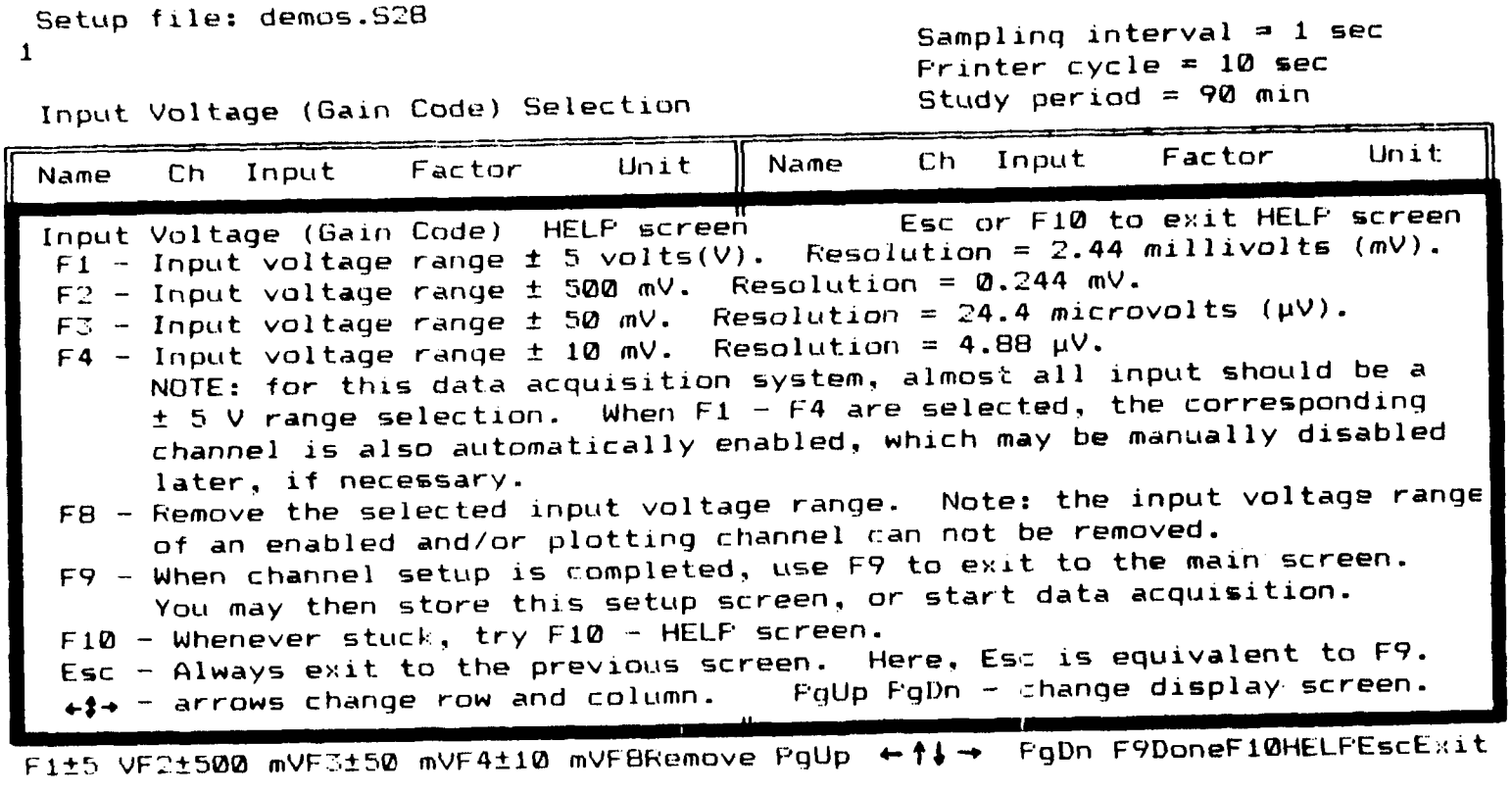


Figure 47 Input Voltage Range and Gain Code Select submodule HELP screen

Conversion Factor Submodule <ConFactor>

Figure 17 shows a more detailed flow chart of the Conversion Factor submodule. Figure 48 is an example of the Conversion Factor submodule screen. "Conversion Factor Selection" appears in the Prompt screen. The "Factor" column heading is highlighted and the current active factor designation starts to blink. If a conversion factor already exists, then the conversion coefficients are displayed in the top row of the Prompt screen. Some long coefficient sets may require two rows to display. The F-key row shows the available options.

Input data often require some processing to reach the final recognizable form/unit. This final unit is usually that entered under the Unit column. For example, a recognizable temperature in °C from a thermocouple must be converted from its initial input form of millivoltage. Four types of data processing methods are provided. Two are specific, for thermocouple and thermistor. The other two are general purpose algorithms, one simple (linear equation conversion), and the other more complex (5th order polynomial conversion).

Remove Conversion Factor F8–Remove. Remove the conversion factor from current active entry. After removal, the channel will have a default unity factor.

Setup Completed or Exit F9/Esc–Done/Exit. This calls the SetupDone submodule to exit the Channel Select module and returns SoS to the Main Selection stage.

Conversion Factor HELP Screens <ConFactorHelp> F10–HELP. Two HELP screens are available (Figure 49).

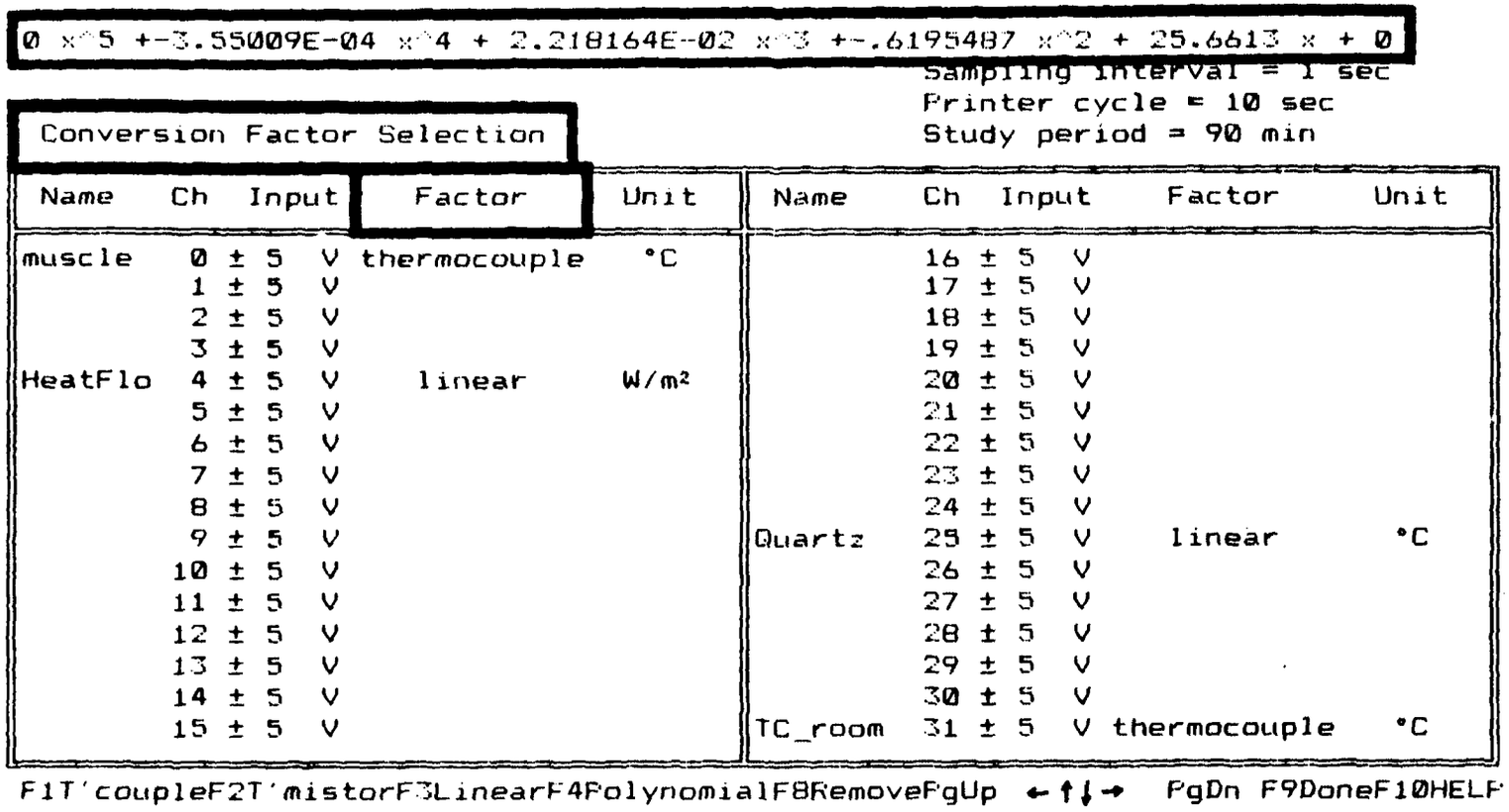


Figure 48 Conversion Factor submodule screen

```
 Setup file: demos.S28
3                                              Sampling interval = 1 sec
                                               Printer cycle = 10 sec
 Conversion Factor Selection                   Study period = 90 min
```

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit

F1 - For thermocouple, a voltage is measured, and then converted to temperature (°C). The conversion coefficients of a Type T (Copper-Constantan) thermocouple equation (from NBS Monograph 125) are automatically installed, if no previously entered coefficients are available. other coefficient sets (e.g. for another thermocouple type) may be entered. A ± 5V input voltage range is also automatically selected.

F2 - For thermistor, a measured resistance (R) in Ω (Ohms) is converted to temperature using T = 1 / [a + b(ln R) + c(ln R)^3] + d {Steinhart-Hart equation}. Coeffieients a, b, c are determined (based on thermistor specification: 2252Ω @ 25°C) and automatically installed. d is an offset value (in °C). If necessary, another set of coefficients may be entered. A ± 5V input voltage range is also automatically selected.

F3 - This linear equation (multiplier a and offset b) converts input data to recognizable units. For example, a measured voltage (x) from an anemometer can be converted to air velocity (m/s) by: a * x + b = m/s.

....MORE.... ↑ or PgDn for next screen Esc or F10 to leave Help screen

F1Thermocouple F2Thermistor F3Linear F4Polynomial F8Remove PgUp ← ↑↓ → PgDn F9Done F10HELP

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit

F4 - For some devices, e.g. thermal anemometers, a simple linear conversion may not be sufficient. More elaborate conversion process is necessary. A 5th order polynomial equation: Ax^5 + Bx^4 + Cx^3 + Dx^2 + Ex + F is provided. A, B, C, D, E, and F are the polymonial coefficients, x represents the measured voltage.

F8 - Remove thermocouple/thermistor/linear/polynomial selection. To change selection, it is not necessary to first remove the old selection. Simply make new selection and enter the new coefficients (and overwrite the old selection and any old coefficients).

F9 - When channel setup is completed, use F9 to exit to the main screen. You will then have the opportunity to store this setup screen, go to another setup, start data acquisition, or save already collected data.

F10 - Whenever stuck, try F10 - HELP screen.

Esc - Always exit to the previous screen. Here, Esc is equivalent to F9.

←↕→ - arrows change row and column. PgUp PgDn - change display screen.

↑ or PgUp for previous screen Esc or F10 to leave Help screen

Figure 49 Conversion Factor submodule HELP screens

Thermocouple F1–T'couple. A thermocouple measures a voltage potential generated between two dissimilar metals. The TC hardware (5B37 module) amplifies the voltage to between 0 and +5V. This voltage must then be converted to temperature data.

The TC 5B37-T-03 module is capable of measuring from -100°C to +400°C, and output a signal between 0 and +5V.

TC module (5B37-T-03) input range: -100°C ----> +400°C

T-type TC voltage at -100 and 400°C
(from NBS monograph 125 Table A8-1-1) E_{mV}: -3.378 mV ----> +20.869 mV

Shift the range of E_{mV} to get a lower limit of zero

$E_{mV+} = E_{mV} + 3.378$ E_{mV+}: 0 ----> +24.247 mV

Convert to range between 0 and +5V

$E_{5V} = E_{mV+} * 206.2$ E_{5V}: 0 ----> +5.000 V

Hence, combining the two equations above

$$E_{mV} = \frac{E_{5V}}{206.2} - 3.378 \qquad \{1\}$$

In Equation {1}, E_{5V} is the voltage that the AC128 measured from the TC input module, whereas E_{mV} is the true thermocouple millivolt signal. Thus, when AC128 measures a voltage between 0 and +5V (E_{5V}), the true millivolt signal generated by the thermocouple (E_{mV}) can be obtained using Equation {1}.

Now, the millivoltage TC signal must be changed tc a temperature data. To convert E_{mV} to temperature (°C), NBS monograph 125[2] gives the following expansion for thermocouples:

---

[2]U.S. Department of Commerce. Thermocouple Reference Tables Based on The IPTS68 - NBS Monograph 125, p.401. National Bureau of Standards, Washington D.C., 1974.

$$T = a + b * E_{mV} + c * E_{mV}^2 + d * E_{mV}^3 + e * E_{mV}^4 + f * E_{mV}^5 \quad \{2\}$$

For the standard T-type copper-constantan thermocouple, the coefficients are (in mV form)

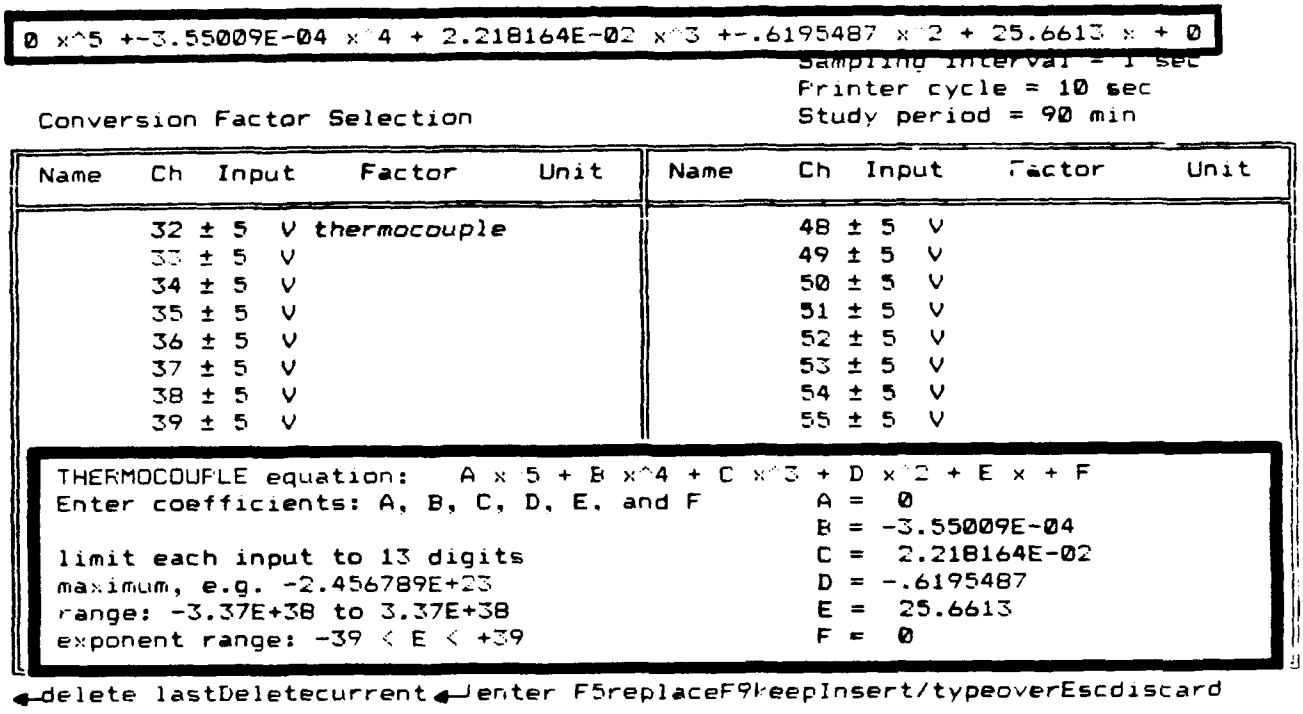
$a = 0$
$b = 25.66$
$c = -0.6196$
$d = 2.218*10^{-2}$
$e = -3.550*10^{-4}$
$f = 0$

Note, the NBS monograph 125 actually gives the conversion equation in the microvolt TC signal form. Equation {2} and the coefficient set have been modified to accommodate the millivolt format. Equation {2} is automatically installed by the SoS. If no other coefficient set has been entered, the T-type TC coefficients are also automatically installed. For other types of TC, different coefficient sets may be manually entered by the operator.

Figure 50 (top) shows a TC selection dialog box. If a TC coefficient set already exists, it is shown in the dialog box and in the Prompt screen. If no previous coefficient exists, then the standard T-type TC coefficient set is shown. The ranges of allowable coefficients are displayed. If a out-of-range number is entered, the SoS will beep and issue a warning message in the dialog box stating the cause of error, and ask the operator to try again.

As new coefficients are entered, the old coefficients are replaced in the dialog box. However, the Prompt screen will still show the old coefficient set until the new one is selected by the operator (by pressing the ↵ Enter key). This allows the operator to retain a view of the old coefficients while the new set is being entered. The operator may want to selectively change one or more of the coefficients, and then use the F9 key to keep the changes and also retain all other unchanged items.

The operator can also replace the current active entry factor with the conversion factor from another channel. F5 is the replacement option key. The SoS will open a replacement window inside the TC Factor entry dialog box (Figure 50, bottom) to ask for the replacement channel number. Afterwards, both channel will possess the exact same TC conversion coefficients. This option is very useful, as often more than one of the same type of TC are employed.



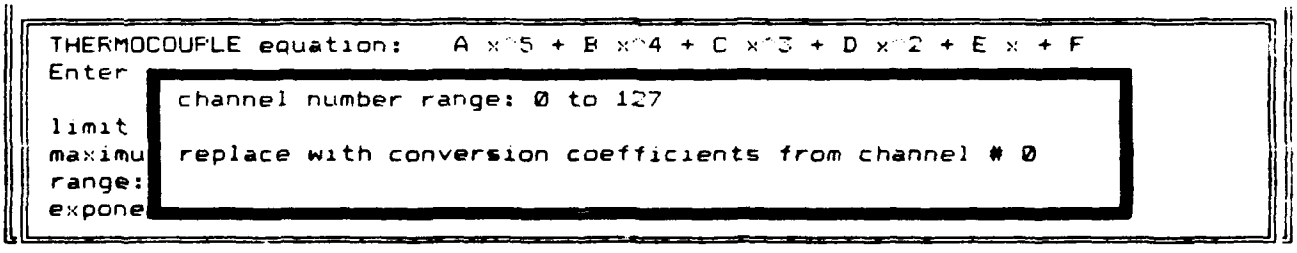


Figure 50 Thermocouple conversion factor entry dialog box and thermocouple conversion factor replacement window

Thermistor F2–T'mistor. Thermistors are usually made of semiconductor material, the resistance of which varies as a function of temperature. This resistance is transformed to a voltage using the bridge circuit incorporated onboard the TM board. The Steinhart-Hart thermistor equation is then used to convert the voltage to temperature data. Equation {3} gives the Steinhart-Hart equation.

$$\frac{1}{T} = a + b * \ln R + c * (\ln R)^3 \qquad \{3\}$$

where a, b, and c are coefficients, and R is resistance value in Ω. The coefficients may be obtained by substituting into Equation {3} three known points of temperature and resistance, and solving the three simultaneous equations. In the AC128 SoS, a slightly modified form is used.

$$T = 1 / [a + b * \ln R + c * (\ln R)^3] + d \qquad \{4\}$$

The coefficient d is an offset that adds one more degree of freedom to the conversion equation.

Equation {4} is automatically installed. If no other coefficient set has been entered previously, then the following coefficient set is automatically selected.

$$a = 1.45937*10^{-3}$$
$$b = 2.39735*10^{-4}$$
$$c = 9.47611*10^{-8}$$
$$d = 0$$

These coefficients are computed based on a thermistor specification of 2252Ω at 25°C (YSI 44033). For thermistors with different specifications, different coefficient sets can be manually entered by the operator.

Figure 51 shows the TM page. The ±5V range and the TM input are automatically selected. The default TM coefficients are also automatically installed and shown in the Prompt screen. Figure 52 (top) shows a TM selection dialog box. The TM coefficient set is shown in the dialog box and in the Prompt screen. The ranges of the allowable coefficients are displayed. If a out-of-range number is entered, the SoS will beep and issue a warning message in the dialog box stating the cause of error, and ask the operator to try again.

As new coefficients are entered, the old coefficients are replaced in the dialog box. However, the Prompt screen will still show the old coefficient set until the new one is selected by the operator (by pressing the ↵ Enter key). This allows the operator to retain a view of the old coefficients while the new set is being entered. The operator may want to selectively change one or more of the coefficients, and then use the F9 key to keep the changes and also retain all other unchanged items.

The operator can also replace the current active entry factor with the conversion factor from another channel. F5 is the replacement option key. The SoS will open a replacement window inside the TM Factor entry dialog box (Figure 52, bottom) to ask for the replacement channel number. Afterwards, both channels will possess the exact same TM conversion coefficients.

Recall that thermistors can only be connected to the TM boards. The SoS will issue a warning message "Thermistor must use channels 96 - 127" in the Prompt screen, if it detects that the operator tries to setup a TM on a nonthermistor channel (Figure 53, top). Conversely, if a TM channel is being setup for nonthermistor input, the SoS will also issue a warning. In Figure 53, (bottom), for example, the warning "channel 96 - 127 are usually thermistor channels" is issued when the SoS detects that a TM channel is being setup for Linear conversion operation. These warnings do not prohibit the operator's actions, they only serve as reminders.

T = 1 / [1.45937E-03 + 2.29735E-04 * (lnR) + 9.47611E-08 * (lnR)^3] + 0

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

Conversion Factor Selection

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
rectal	96	± 5 V	thermistor	°C	rectal	112	± 5 V	thermistor	°C
	97	± 5 V	thermistor	°C		113	± 5 V	thermistor	°C
	98	± 5 V	thermistor	°C		114	± 5 V	thermistor	°C
	99	± 5 V	thermistor	°C		115	± 5 V	thermistor	°C
	100	± 5 V	thermistor	°C		116	± 5 V	thermistor	°C
	101	± 5 V	thermistor	°C		117	± 5 V	thermistor	°C
	102	± 5 V	thermistor	°C		118	± 5 V	thermistor	°C
	103	± 5 V	thermistor	°C		119	± 5 V	thermistor	°C
	104	± 5 V	thermistor	°C	TM-room	120	± 5 V	thermistor	°C
	105	± 5 V	thermistor	°C		121	± 5 V	thermistor	°C
	106	± 5 V	thermistor	°C		122	± 5 V	thermistor	°C
TM-bath	107	± 5 V	thermistor	°C		123	± 5 V	thermistor	°C
	108	± 5 V	thermistor	°C		124	± 5 V	thermistor	°C
	109	± 5 V	thermistor	°C		125	± 5 V	thermistor	°C
	110	± 5 V	thermistor	°C		126	± 5 V	thermistor	°C
	111	± 5 V	thermistor	°C		127	± 5 V	thermistor	°C

F1T'coupleF2T'mistorF3LinearF4PolynomialF8RemovePgUp ← ↑ ↓ → PgDn F9DoneF10HELP

Figure 51 Sample thermistor channel setup page

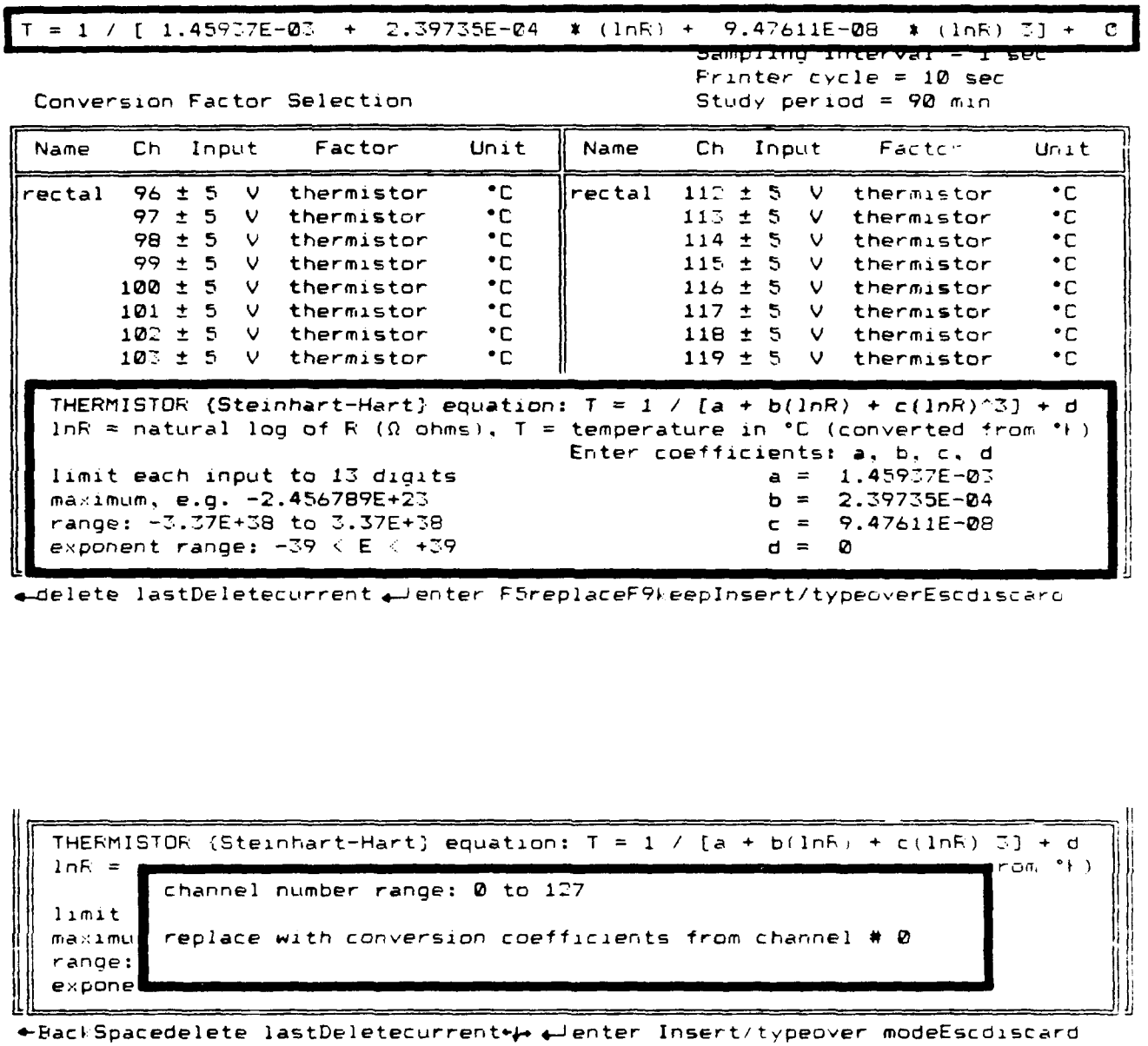
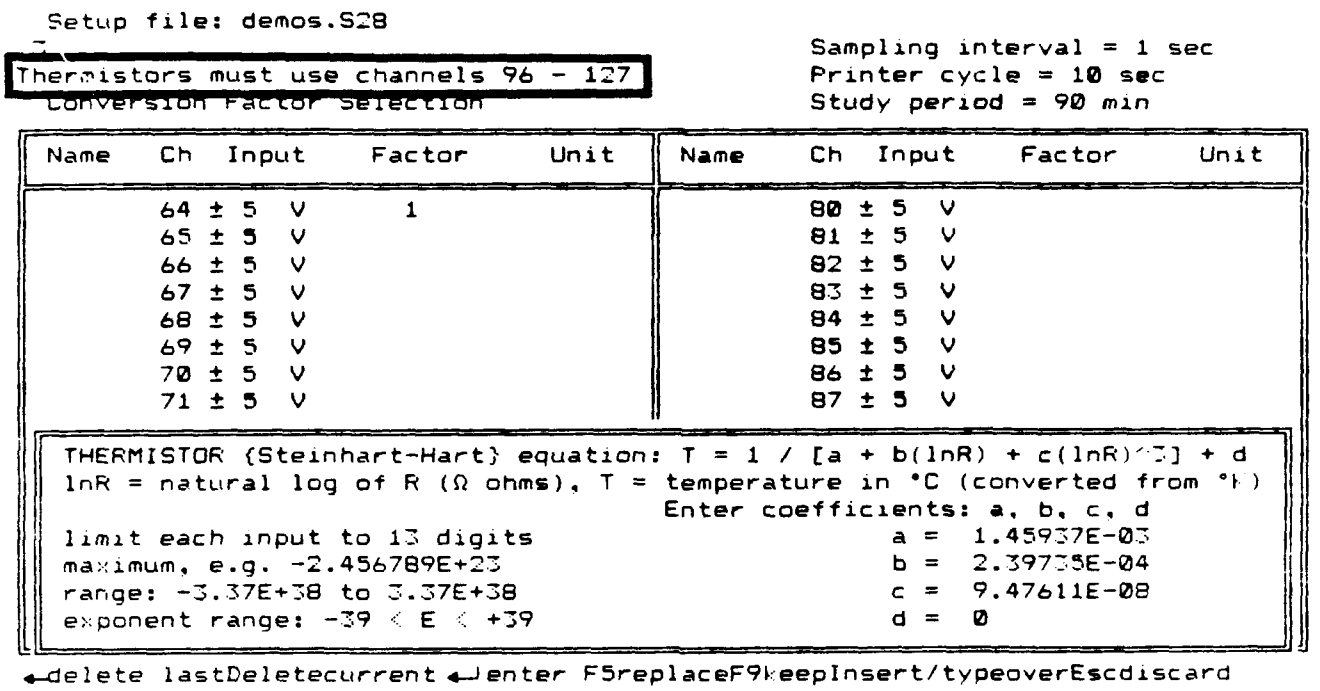


Figure 52 Thermistor conversion factor entry dialog box and thermistor conversion factor replacement window



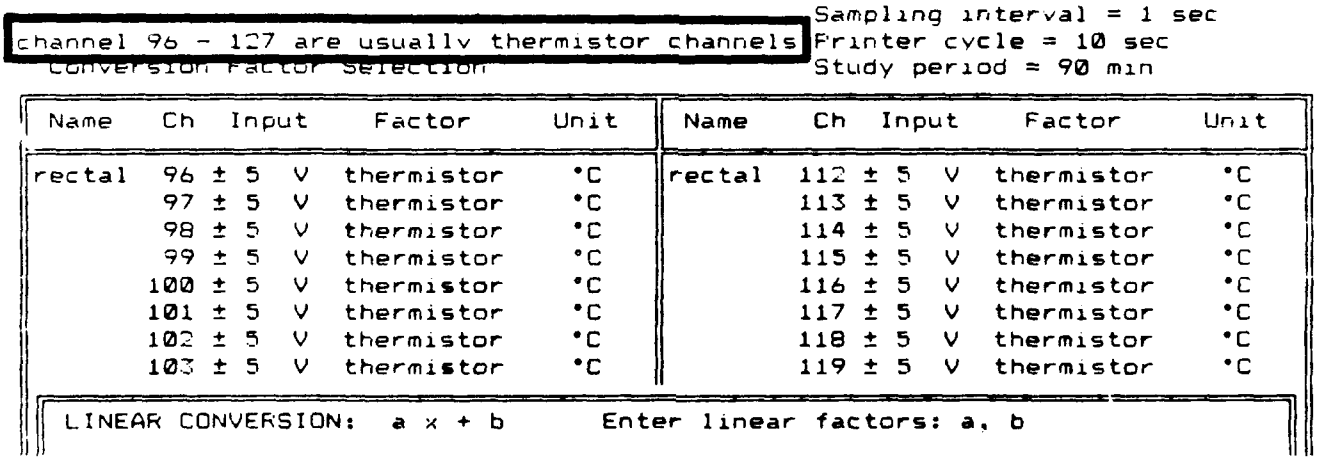


Figure 53 Inappropriate thermistor channel setup warning messages

Linear Conversion F3–Linear. The input data are converted linearly. The linear equation is

$$Y = m * X + b, \quad \{5\}$$

where m is the slope and b is the y-axis intercept value. Y represents the final data form, and X is the raw input data. For instance, when voltage is converted to temperature, Y is temperature in °C, and X is voltage in V or mV.

Figure 54 (top) shows a Linear conversion entry dialog box. If linear factors already exist, they are shown in the dialog box and in the Prompt screen. If no previous factor exists, then the default unity settings of m = 1, and b = 0 are installed. The ranges of the allowable coefficients are displayed. If a out-of-range number is entered, the SoS will beep and issue a warning message in the dialog box stating the cause of error, and ask the operator to try again.

The F9 key allows the operator to change one of the factors and retain the other one unchanged. As new factors are entered, old factors in the dialog box are replaced. However, the Prompt screen will retain the old factors until the new ones are selected by the operator (by pressing the ↵ Enter key). This allows the operator to retain a view of the old factors while the new set is being entered.

F5 is the replacement option. The SoS will open a replacement window inside the Linear Factor entry dialog box (Figure 54, bottom) to ask for the replacement channel number. Afterwards, both channels will possess the exact same linear conversion factors.

Linear Conversion F3–Linear. The input data are converted linearly. The linear equation is

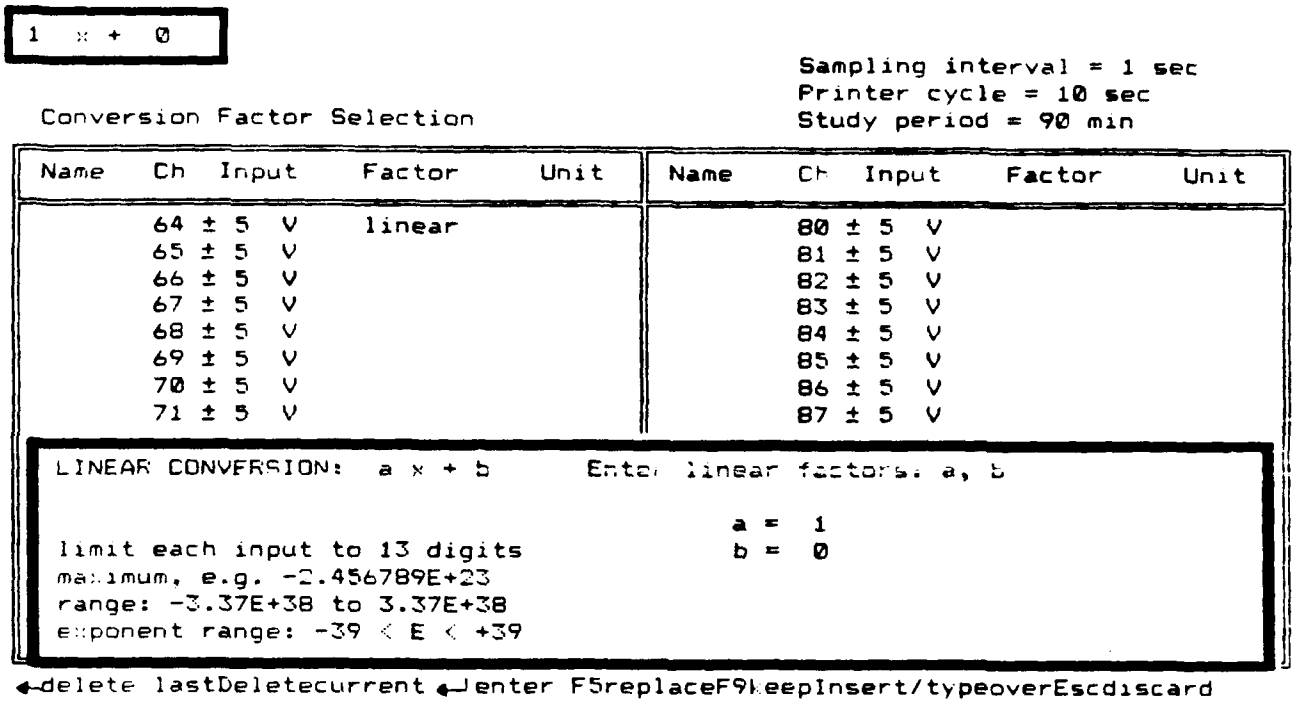
$$Y = m * X + b, \qquad \{5\}$$

where m is the slope and b is the y-axis intercept value. Y represents the final data form, and X is the raw input data. For instance, when voltage is converted to temperature, Y is temperature in °C, and X is voltage in V or mV.

Figure 54 (top) shows a Linear conversion entry dialog box. If linear factors already exist, they are shown in the dialog box and in the Prompt screen. If no previous factor exists, then the default unity settings of m = 1, and b = 0 are installed. The ranges of the allowable coefficients are displayed. If a out-of-range number is entered, the SoS will beep and issue a warning message in the dialog box stating the cause of error, and ask the operator to try again.

The F9 key allows the operator to change one of the factors and retain the other one unchanged. As new factors are entered, old factors in the dialog box are replaced. However, the Prompt screen will retain the old factors until the new ones are selected by the operator (by pressing the ↵ Enter key). This allows the operator to retain a view of the old factors while the new set is being entered.

F5 is the replacement option. The SoS will open a replacement window inside the Linear Factor entry dialog box (Figure 54, bottom) to ask for the replacement channel number. Afterwards, both channels will possess the exact same linear conversion factors.



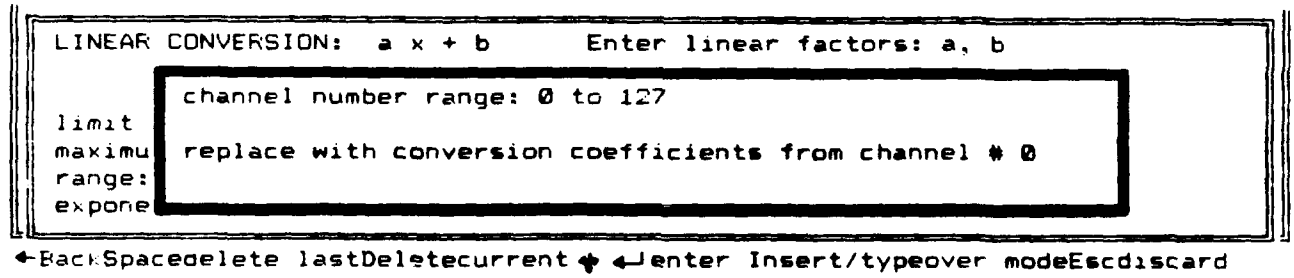


Figure 54 Linear conversion factor entry dialog box and linear conversion factor replacement window

Polynomial Conversion F4–Polynomial. For some devices, for example, thermal anemometers, a simple linear conversion may not be sufficient, and a more elaborate conversion process is necessary. A 5th order polynomial, Equation {6}, is provided.

$$y = A x^5 + B x^4 + C x^3 + D x^2 + E x + F \qquad \{6\}$$

A, B, C, D, E and F are coefficients of the polynomial. The default setting has all coefficients equal to zero.

Figure 55 (top) shows a polynomial entry dialog box. If polynomial factors already exist, they are shown in the dialog box and in the Prompt screen. If no previous factor exists, then all factors are default to zero. The ranges of the allowable coefficients are displayed. If a out-of-range number is entered, the SoS will beep and issue a warning message in the dialog box stating the cause of error, and ask the operator to try again.

The F9 key allows the operator to selectively change one or more of the factors and retain the others unchanged. As new factors are entered, old factors in the dialog box are replaced. However, the Prompt screen will retain the old factors until the new ones are selected by the operator (by pressing the ↵ Enter key). This allows the operator to retain a view of the old polynomial factors while the new set is being entered.

F5 is the replacement option. The SoS will open a replacement window inside the Polynomial Factor entry dialog box (Figure 55, bottom) to ask for the replacement channel number. Afterwards, both channels will possess the exact same polynomial conversion factors.

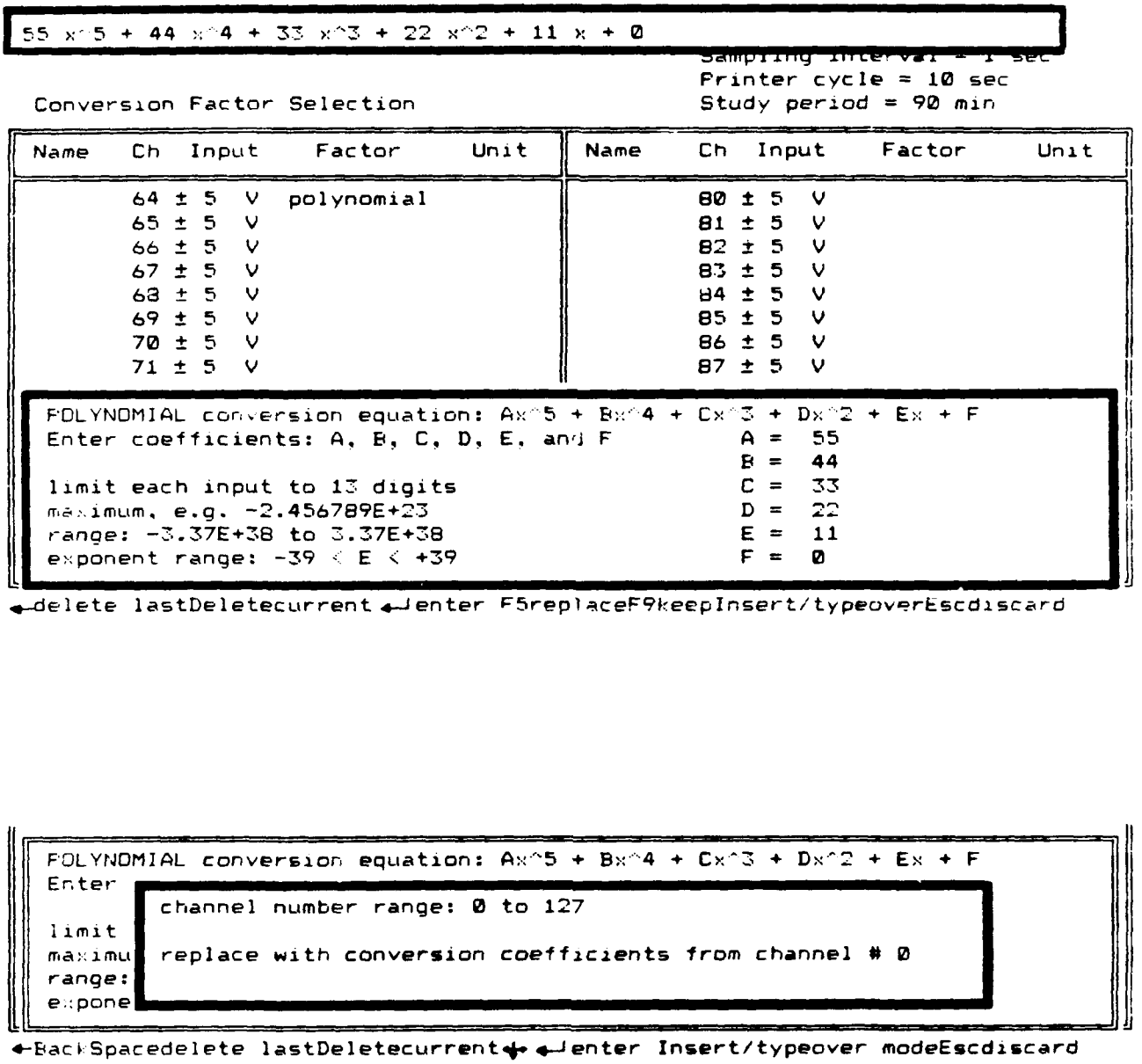


Figure 55 Polynomial conversion factor entry dialog box and polynomial conversion factor replacement window

Unit Label Submodule <UnitLabel>

Figure 56 is an example of the Unit Label screen. "Unit Label Selection" appears in the Prompt screen. The "Unit" column heading is highlighted and the current unit label starts to blink. The F-key row shows the available function key options.

Enter Unit Label F1–Enter Unit. Enter a unit label for the final recognizable data form for this channel (Figure 57, top). If a device name already exists, then it is shown in the entry dialog box. The unit label is a alphanumeric name of a maximum of 7 characters long. If a disallowed character is entered, the SoS will beep and issue an error message in the dialog box indicating which is the offending character. The illegal character is ignored and the operator is prompted to continue with the label entry. The default unit label is "none".

The operator may also replace the current active entry label with the unit label from another channel (by using the F5-replace key). The SoS will open a replacement window inside the Unit entry dialog box (Figure 57, bottom) to ask for the replacement channel number. Afterwards, both channels will have the exact same unit label. This option is useful if several data channels have the same type of device.

Remove Unit Label F8–Remove Unit. Remove the unit label from this data channel. The unit label will be replaced with "none".

Setup Completed or Exit F9/Esc–Done/Exit. This calls the SetupDone submodule to exit the Channel Select module and returns SoS to the Main Selection stage.

Unit Label HELP Screen <UnitLabelHelp> F10–HELP. See Figure 58. A portion of this HELP screen <NameUnitHELP1> is shared with the Device Name HELP screen.

Setup file: demos.S28

4

Sampling interval = 1 sec
Printer cycle = 10 sec
Study period = 90 min

Unit Label Selection

Name	Ch	Input	Factor	Unit	Name	Ch	Input	Factor	Unit
muscle	0	± 5 V	thermocouple	°C		16	± 5 V		
	1	± 5 V				17	± 5 V		
	2	± 5 V				18	± 5 V		
	3	± 5 V				19	± 5 V		
HeatFlo	4	± 5 V	linear	W/m²		20	± 5 V		
	5	± 5 V				21	± 5 V		
	6	± 5 V				22	± 5 V		
	7	± 5 V				23	± 5 V		
	8	± 5 V				24	± 5 V		
	9	± 5 V			Quartz	25	± 5 V	linear	°C
	10	± 5 V				26	± 5 V		
	11	± 5 V				27	± 5 V		
	12	± 5 V				28	± 5 V		
	13	± 5 V				29	± 5 V		
	14	± 5 V				30	± 5 V		
	15	± 5 V			TC_room	31	± 5 V	thermocouple	°C

F1Enter Unit F8Remove Unit PgUp ← ↑ ↓ → PgDn F9Done F10HELP EscExit

Figure 56 Unit Label submodule screen

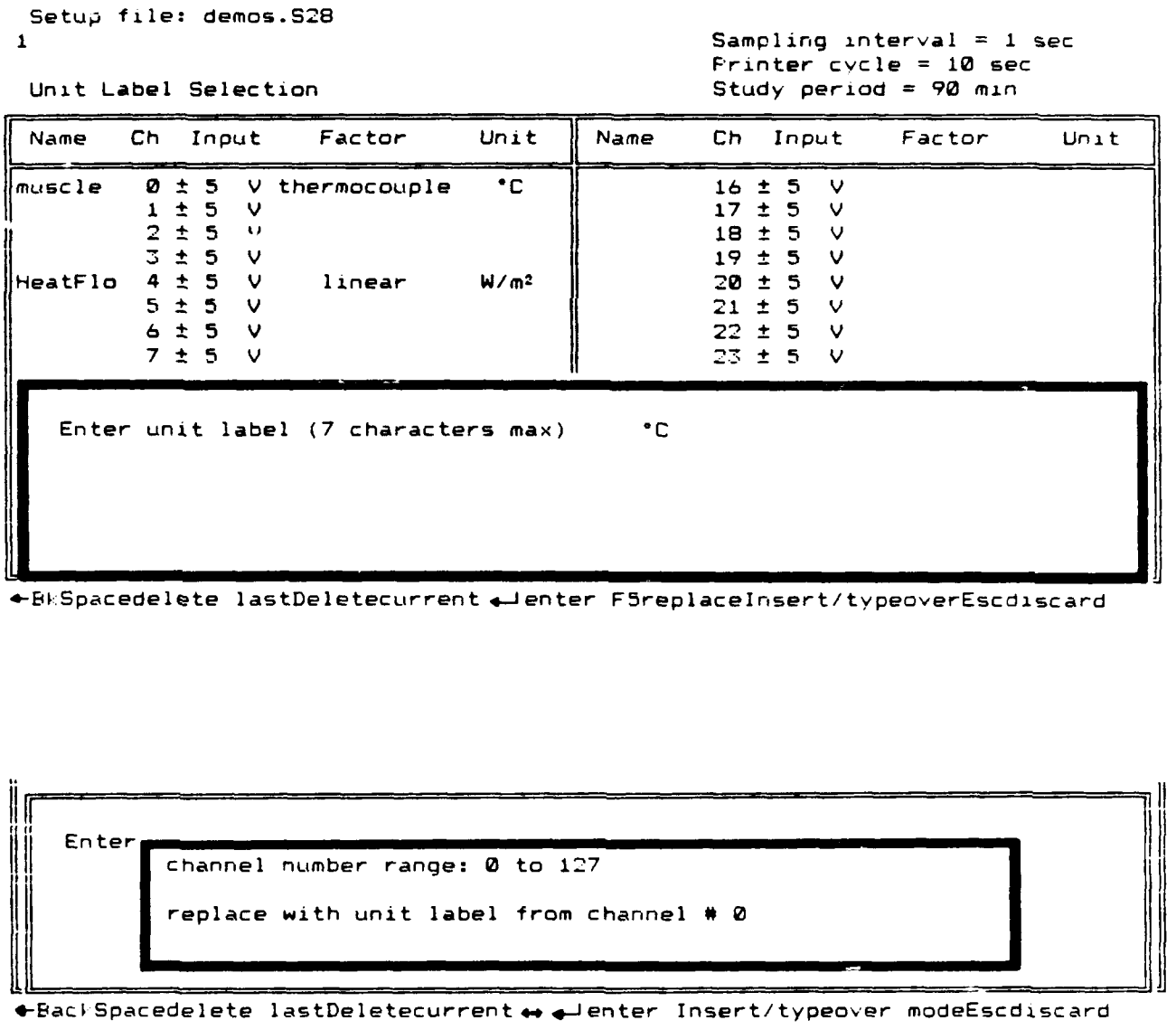


Figure 57 Unit Label entry dialog box and Unit Label replacement window

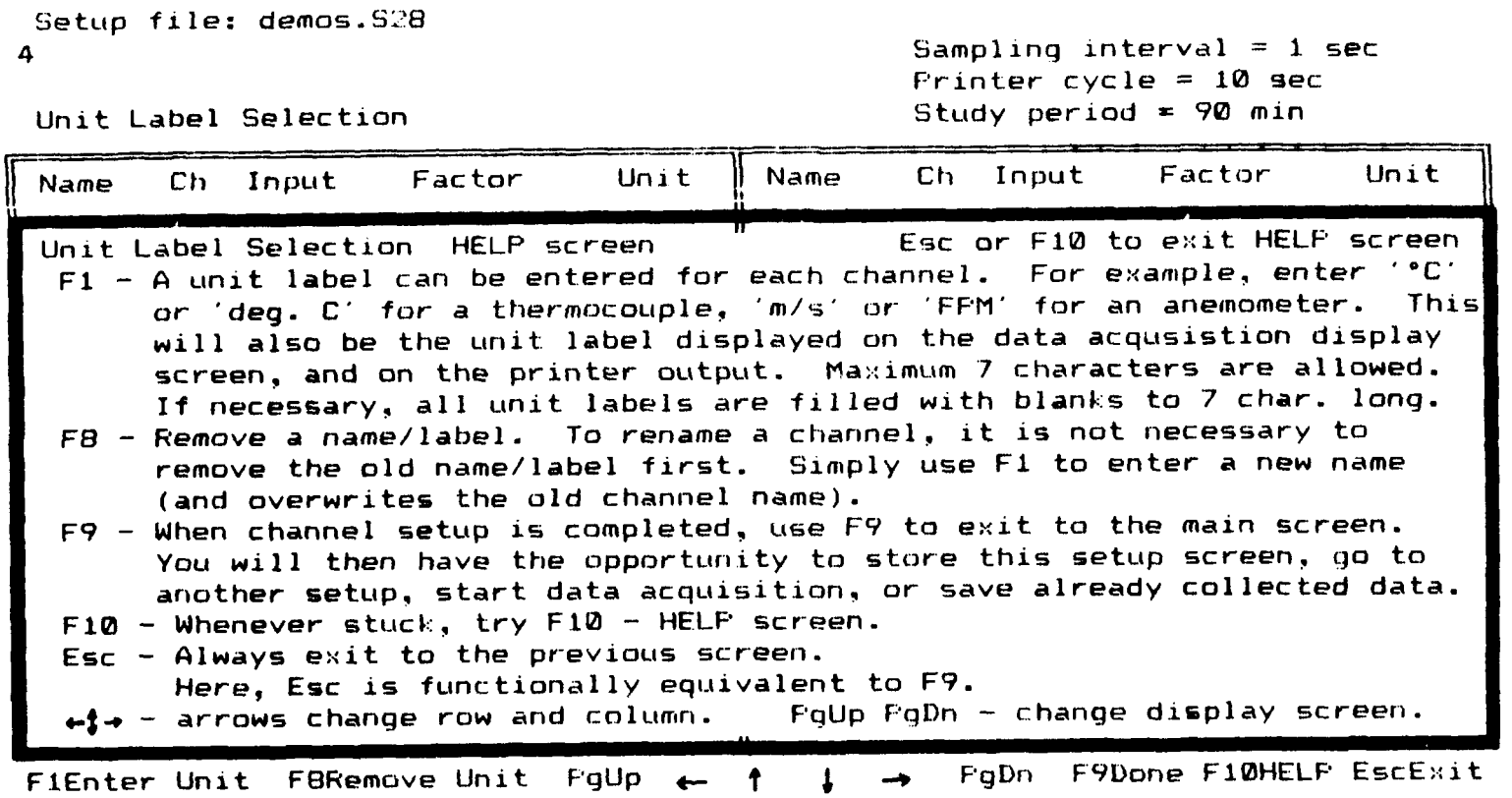


Figure 58 Unit Label submodule HELP screen

DATA ACQUISITION <DataAcq>

The Data Acquisition module immediately checks to see if the Timing parameters have been entered. If no acquisition Timing has been entered, the appropriate error message is issued in the Timing screen (Figure 59) and the SoS is returned to the Main Selection stage. The error message also instructs the operator to use the F4-Modify key (at the Main Selection stage) to get into the Channel Select module to set the Timing parameters.

If Timing parameters are present then the operator will be asked to enter a study identification remark (Figure 60), and a storage file name for the data to be collected (Figure 61). Extensive on-screen instructions are provided. Error in entry will be detected and the cause of error displayed on screen. The storage filename must be entered whereas the study identification remark is optional. Both the study identification and the data filename will be included in the data storage file and printed on the printer output.

Figure 62 shows the initial Data Acquisition screen. No value is displayed since data acquisition has not yet commenced. The data filename is shown in the Prompt screen. The Name, Ch, and Unit column headings remain unchanged. The previous Input and Factor columns now display the instantaneous Data and the Average data. The Average data are also that printed on the printer output. If data plotting has been selected then the plotting screens are also cleared. Note, because the plotting screens are cleared at this stage, therefore, any data plotted during a previous study run are available for viewing, or printing (using Print Screen) up to this point.

Figure 18 shows the Data Acquisition module flow chart.

Select/Change Printer <DataAcqPrinterSelect> F4-Printer

In the Prompt screen of Figure 63 (top), the selected printer number is displayed. Initially, the SoS assumes that the printer is connected to the PC parallel port 1 (LPT1). The SoS therefore defaults to printer number 1. The printer number may be changed to 2 or 3, if the printer is connected to other parallel ports or to serial ports. Figure 64 shows the printer number selection window. The SoS will accept only printer numbers 1, 2 and 3, as allowed by PC. Alternatively, the printer may be disabled by entering printer number 0, the Prompt screen will show "printer disabled" (Figure 63, bottom). No printer hardcopy record will then be produced.

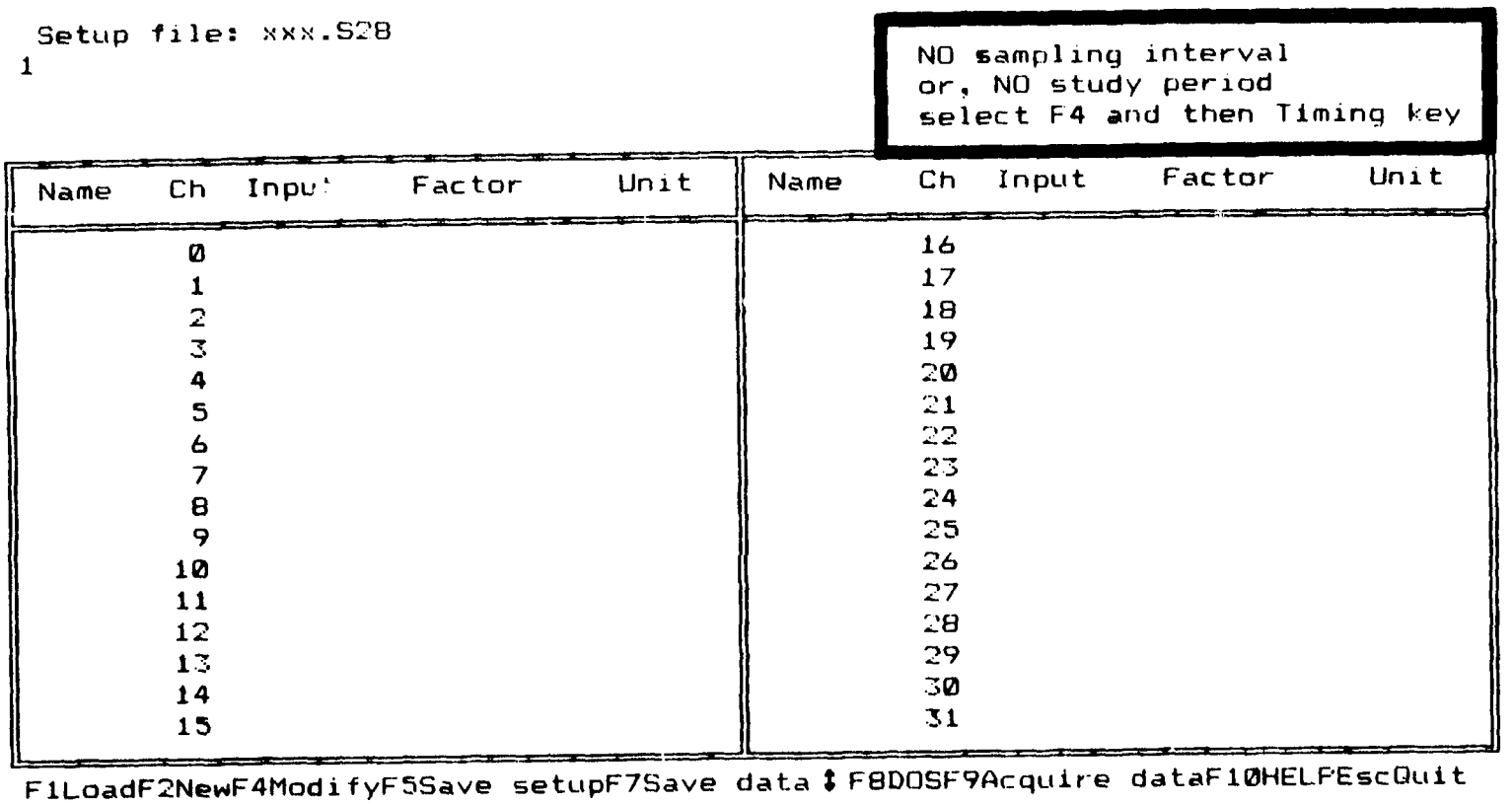


Figure 59 No data acquisition Timing parameter warning message

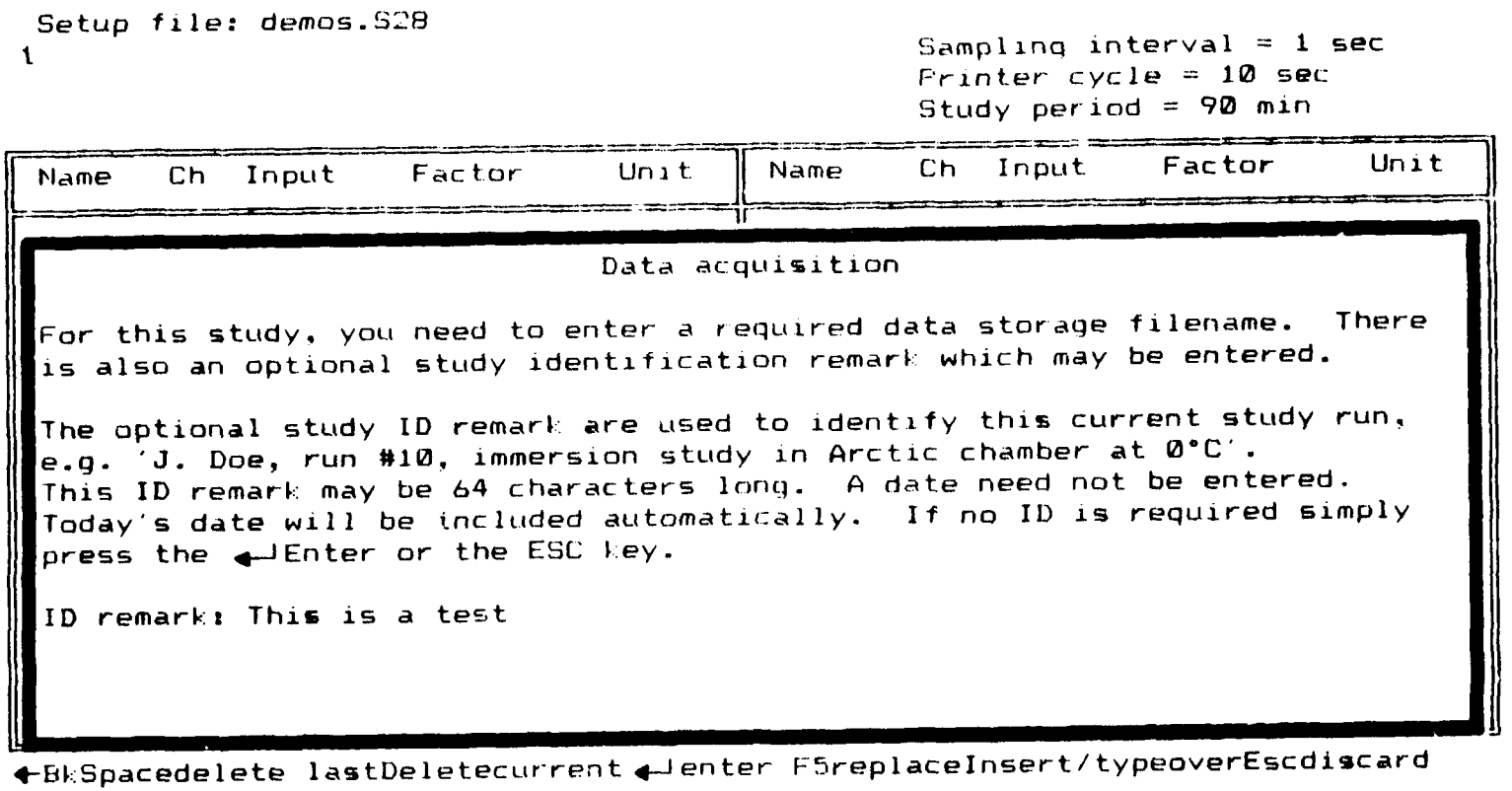


Figure 60 Data Acquisition study identification remark entry dialog box

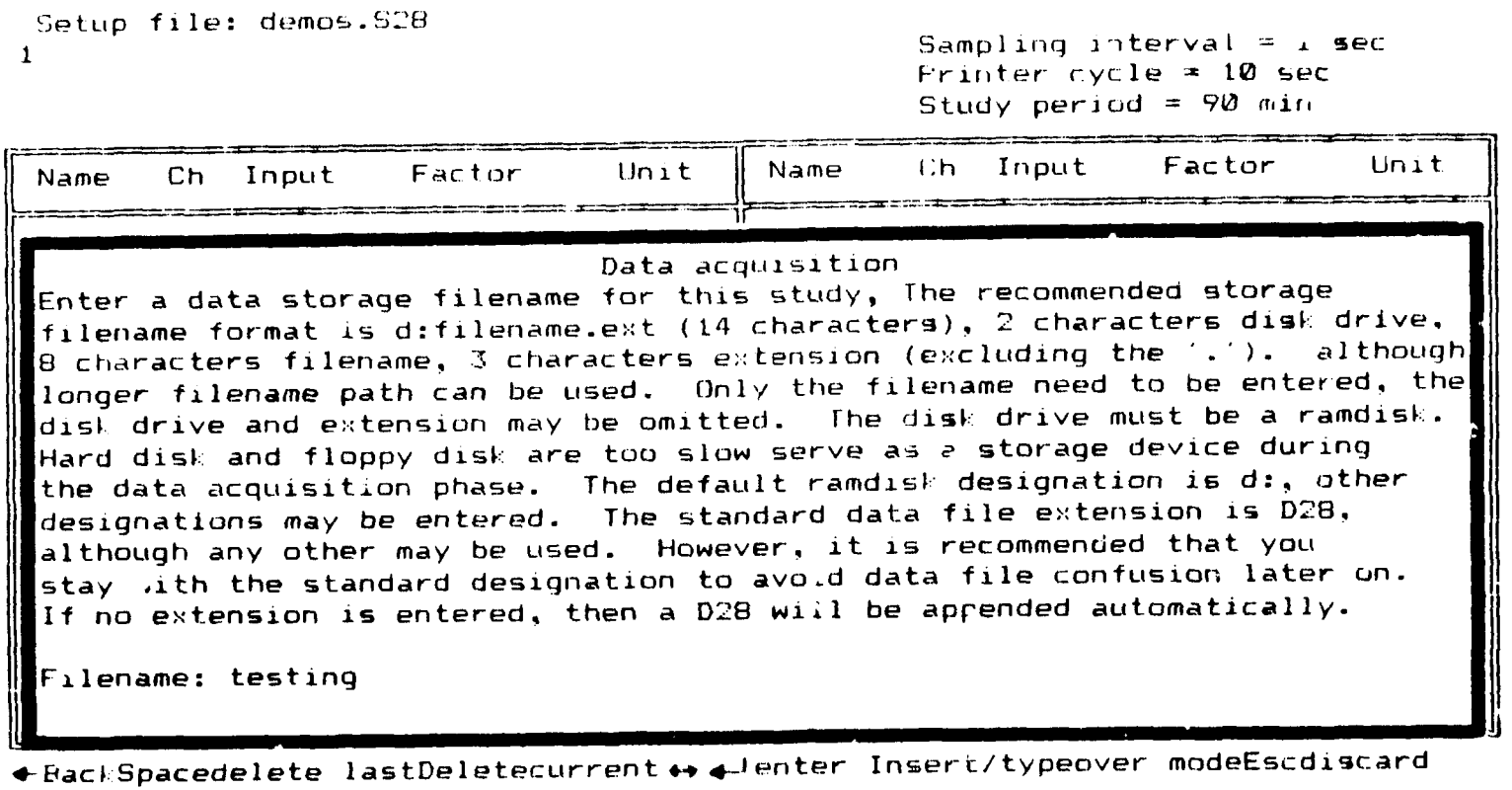


Figure 61 Data Acquisition data filename entry dialog box

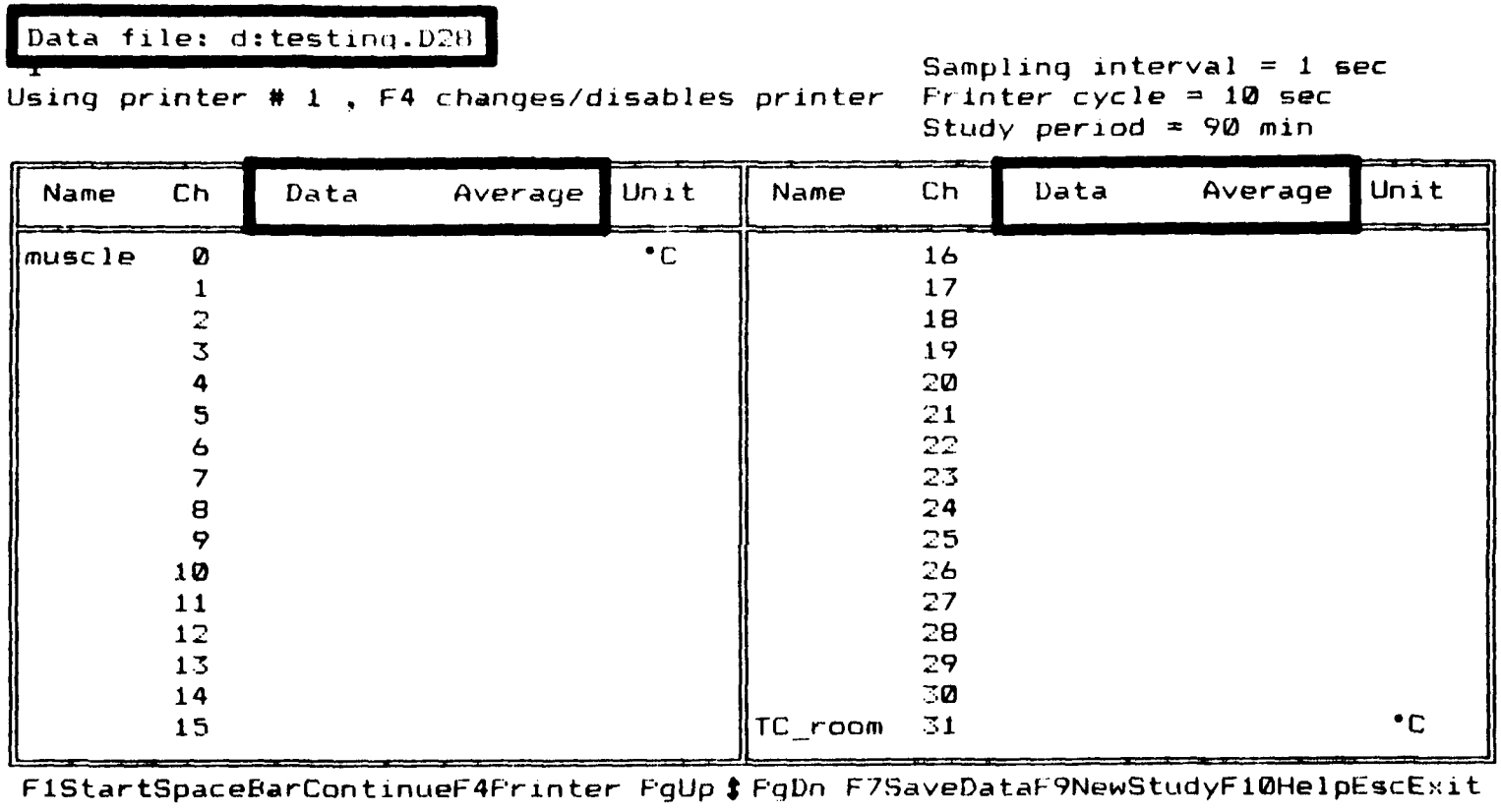
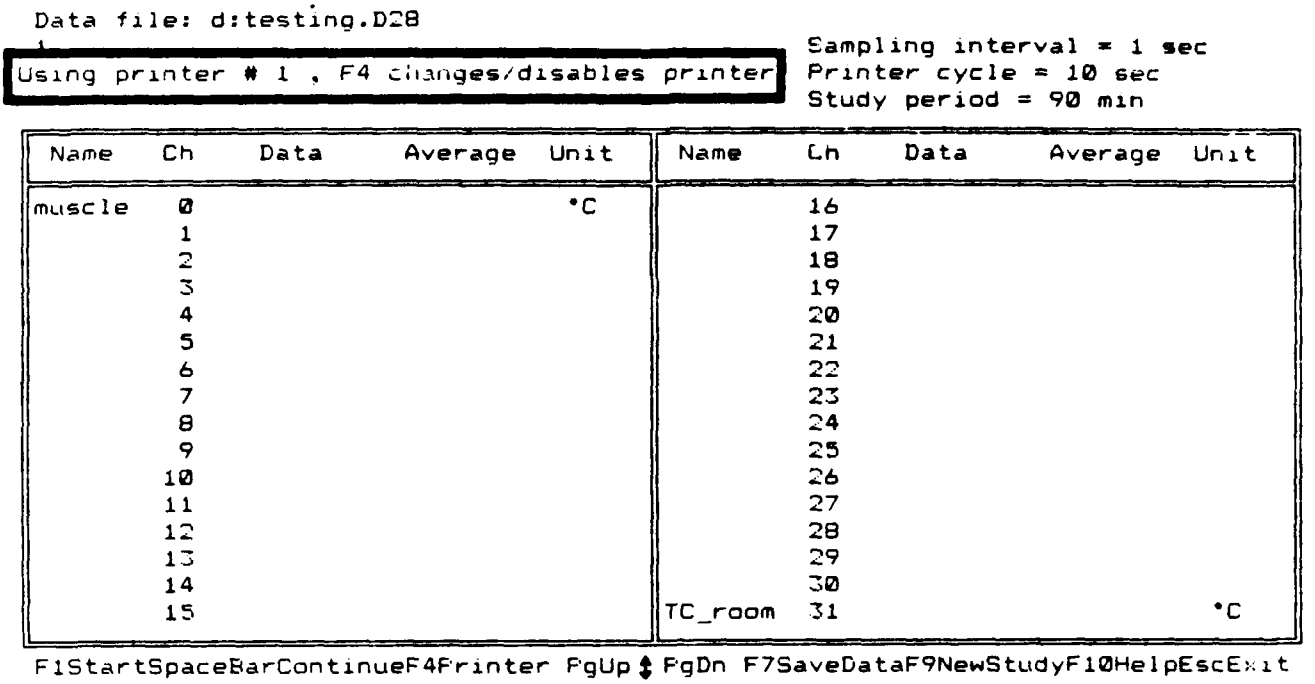


Figure 62 Data Acquisition module initial screen (without data)



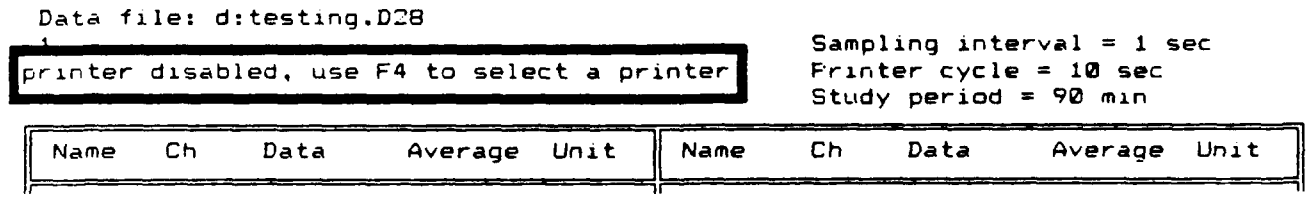


Figure 63 Printer status messages

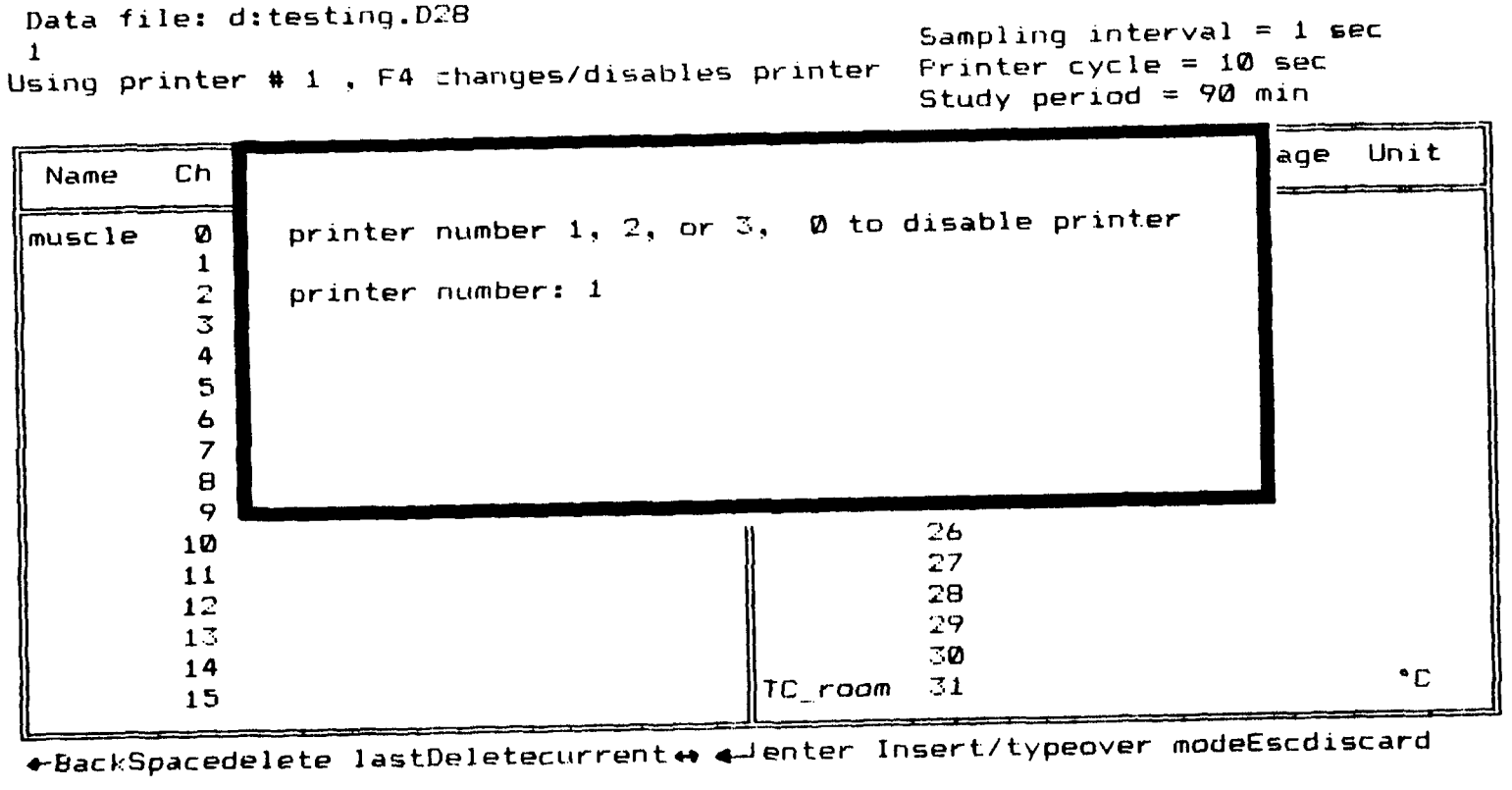


Figure 64 Printer selection dialog box

Save Collected Data <SaveData> F7-SaveData

Saves a permanent copy of data collected during data acquisition. This option was also available at the Main Selection stage, and the same submodule <SaveData> is called here to perform the task. Recall, data collected during data acquisition are temporarily retained on a ramdisk, which unless saved onto a more permanent medium, cannot be recovered once the PC power is switched off. Therefore, the collected data should be saved at the earliest opportunity. Usually, the save will be to a hard disk or a floppy disk, although any logical device connected to the PC may be used. The Prompt screen will display status of a successful Save, such as "1 File copied" or error message of an unsuccessful Save, such as "File not found" (see Figure 31).

A partial data set may also be saved. When the data acquisition is temporarily suspended (as will be described), this <SaveData> option is available. A permanent copy of the data collected up to the point of suspension may be saved.

Start A New Study <DataAcqEnd> F9-NewStudy

Prepare for a new study by restoring the display screens and return to Main Selection module. The SoS will display a "Restoring setup screen" message window (Figure 65) until the restoring action is completed. All four channel setup screens are restored. The plotting screens are not erased, therefore, the data plots are available for viewing and hardcopy output (using Print Screen) when the SoS returns to the Main Selection stage.

Exit Data Acquisition <DataAcqEnd> ESC-Exit

Exit the current data acquisition run. This selection has the same effect as the New Study (F9 key) option.

Data Acquisition HELP Screen <DataAcqHelp> F10-HELP

See Figure 66.

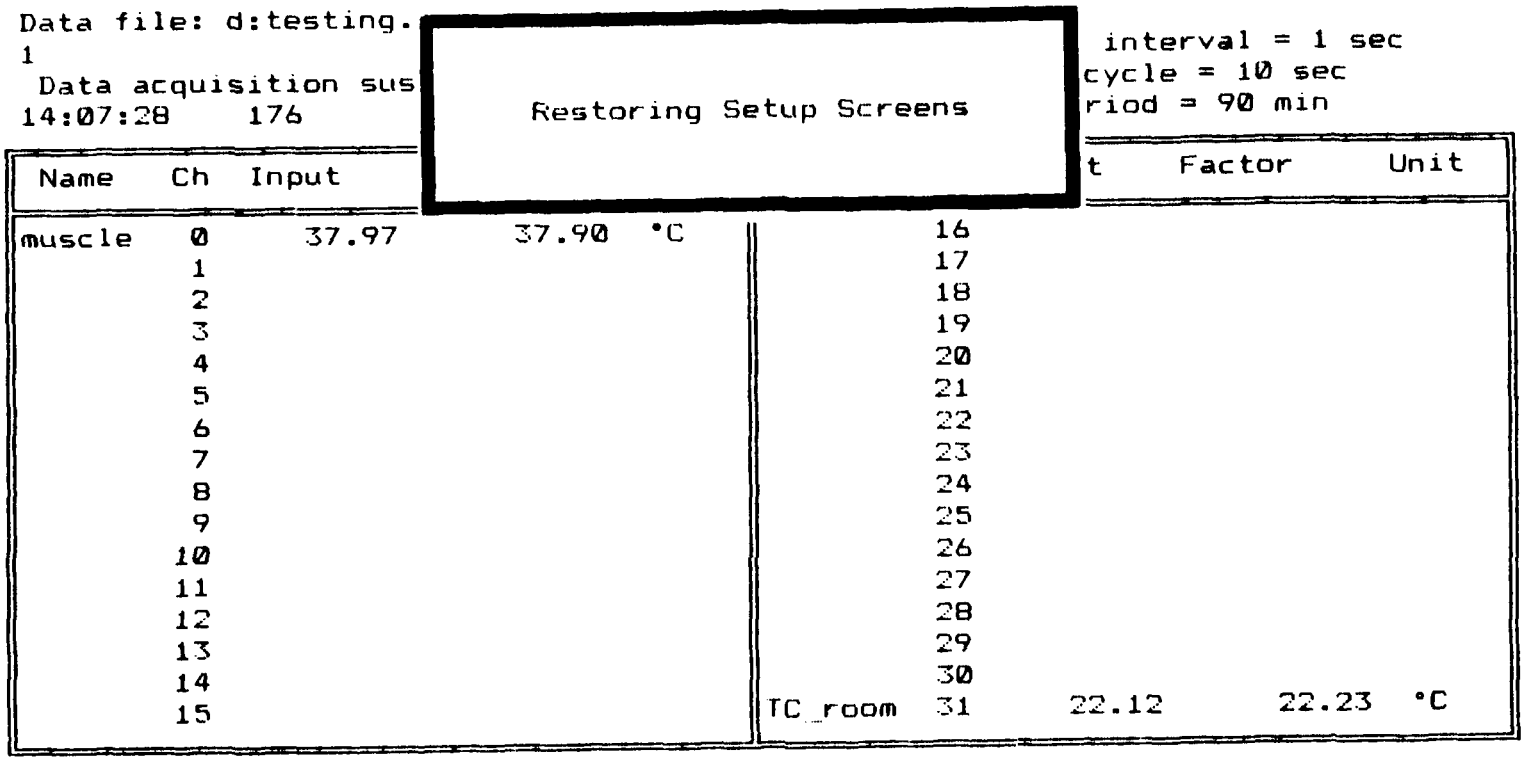


Figure 65 Restoring setup screens message window

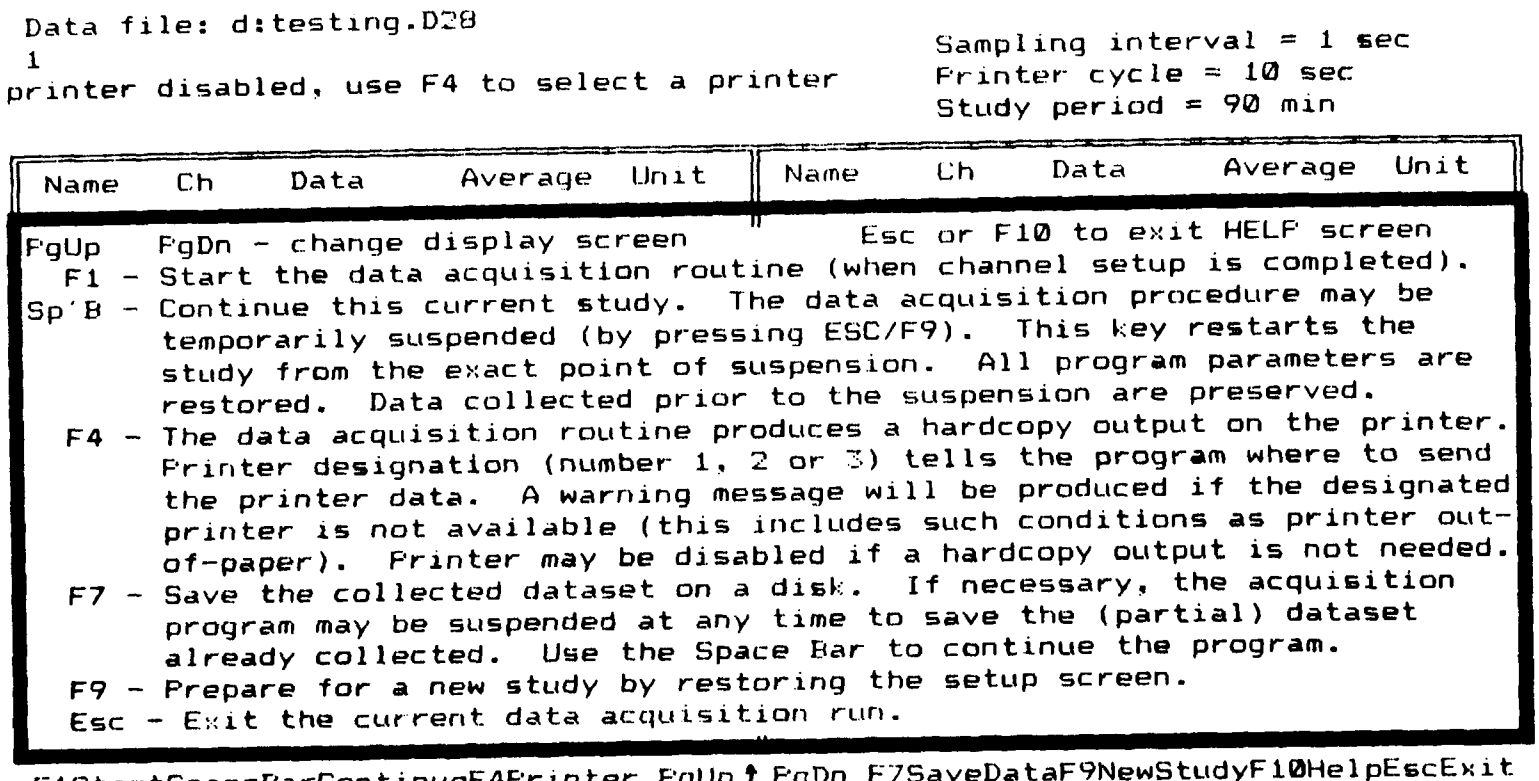


Figure 66 Data Acquisition module HELP screen

Start Data Collection <DataAcq3> F1-Start/SpaceBar-Continue

Data will be collected once every sampling period from each one of the enabled channels. The heart of the data collection algorithms is an Assembly language subroutine (<Acq> in <DataAcqSample>). <Acq> serves two roles: 1) addresses the A/D board control register commanding it to collect data, and 2) accesses the content of the A/D board data registers, and sends the data back to the SoS.

To reduce electrical noise, multiple samples of raw data are actually collected during each sampling cycle. For data collected on the GP boards, 33 samples of raw data are collected then averaged. This averaged datum is then send back to the SoS as a single raw datum. For the TM boards, 66 samples are averaged. The numbers 33 and 66 are chosen to minimize the 60Hz noise.

Upon receiving the raw data, the SoS processes the data in accordance with the specified conversion factors, if any. If no conversion factor was specified, then a unity factor is assumed. The thermocouple submodule <DataAcqThermocouple> processes the raw data according to Equations {2} and {3}. Linear and polynomial conversions use Equations {5} and {6}, respectively.

The thermistor submodule <DataAcqThermistor> not only processes the data according to Equation {4}, it also contains a 2-pole low-pass recursive digital filter. The TM boards do not contain any hardware filtering, and a low-pass filter is necessary to reduce the electrical noises that are picked up by the TM boards and the thermistor cables. A 2-pole low-pass filter is programmed into the thermistor submodule. Equation {7} gives the filter equation.

$$g(n) = 0.3\ f(n) + 0.5\ g(n-1) + 0.2\ g(n-2) \qquad \{7\}$$

The raw data are first filtered with Equation {7} before being processed by the Steinhart-Hart Equation {4}.

The conversion factor processed data are displayed on the monitor screens under the "Data" columns. These data are also stored in the PC on the ramdisk. The data are stored in ASCII form, hence are directly accessible by commercially available post processing tools such as spread sheet programs and statistical packages. Figure 67 shows a sample of the ramdisk data. The AC128 DAS version and date is included. Data filename and study identification remark (if entered) also precede the data listing.

Biophysics Data Acquisition System ac128.20 900807 Stephen KW. Chang
08-07-1990 d:testing.D28 This is a test
14:34:21 38.08261 22.84734
14:34:22 38.02253 22.84734
14:34:23 37.95385 22.8738
14:34:24 38.03111 22.83854
14:34:25 37.99677 22.91787
14:34:26 37.96243 23.00601
14:34:27 38.00536 22.85617
14:34:28 37.90234 22.86497
14:34:29 38.05685 22.80327
14:34:30 38.02253 22.78563
14:34:31 38.07402 22.79445
14:34:32 37.95385 22.73272
14:34:33 37.9281 22.74154
14:34:34 38.00536 22.90904
14:34:35 37.97102 22.84734


14:34:36

14:34:36 37.93668 22.768
14:34:37 38.01394 22.82971
14:34:38 37.9796 22.97076
14:34:39 37.91951 22.82971
14:34:40 38.06545 22.8209
14:34:41 38.00536 22.8209
14:34:42 37.89375 22.65335
14:34:43 37.96243 22.75918
14:34:44 37.87658 22.73272
14:34:45 37.97102 22.75036
14:34:46 37.96243 22.83854
14:34:47 37.96243 22.65335
14:34:48 38.02253 22.83854
14:34:49 37.88516 22.768
14:34:50 38.05685 22.68863
14:34:51 37.98819 22.70627

Figure 67 Sample ramdisk data

At the printer output cycle time, the data are averaged and displayed on the screens under the "Average" columns. If data plotting was designated, then the averaged data from the designated channel are plotted. Also, the averaged data are printed on the printer for a hardcopy record, if printing is enabled. Figure 68 shows a sample of the printer output. Date, data filename and study identification remark (if entered) precedes the data. The data are delineated by the acquisition time. Each line of data is organized in this format: device name, channel number, data, unit label.

Each time before printing, the SoS checks the status of the printer. A warning message is displayed if the printer is out-of-paper (Figure 69, top), or for any reason unavailable (Figure 69, bottom). Upon detection of printer error, the data acquisition is temporarily suspended. The operator can choose to correct the printer problem or disable the printer, and then resume data acquisition.

At the end of the study period, a "Data acquisition completed" message will appear in the Prompt screen (Figure 70). The SoS will also chime three times to inform the operator.

During data acquisition, the time-of-day and the data sample number are displayed in the Prompt screen (Figure 71).

```
08-07-1990  d:testing.D28  This is a test
14:33:53
muscle      0        38.02     °C
TC_room    31        22.75     °C
14:34:03
muscle      0        38.02     °C
TC_room    31        22.76     °C
14:34:12
muscle      0        38.05     °C
TC_room    31        22.69     °C
14:34:22
muscle      0        38.01     °C
TC_room    31        22.78     °C
14:34:32
muscle      0        38.00     °C
TC_room    31        22.85     °C

⟹  14:34:36

14:34:42
muscle      0        37.97     °C
TC_room    31        22.82     °C
14:34:52
muscle      0        37.97     °C
TC_room    31        22.73     °C
14:35:02
muscle      0        38.03     °C
TC_room    31        22.88     °C
14:35:12
muscle      0        37.96     °C
TC_room    31        22.92     °C
14:35:22
muscle      0        37.98     °C
TC_room    31        22.91     °C
14:35:32
muscle      0        37.98     °C
TC_room    31        22.66     °C
14:35:41
muscle      0        37.96     °C
TC_room    31        22.55     °C
14:35:51
muscle      0        37.93     °C
TC_room    31        22.50     °C
14:36:01
muscle      0        37.94     °C
TC_room    31        22.63     °C
14:36:11
muscle      0        37.95     °C
TC_room    31        22.65     °C
```

Figure 68 Sample printer output

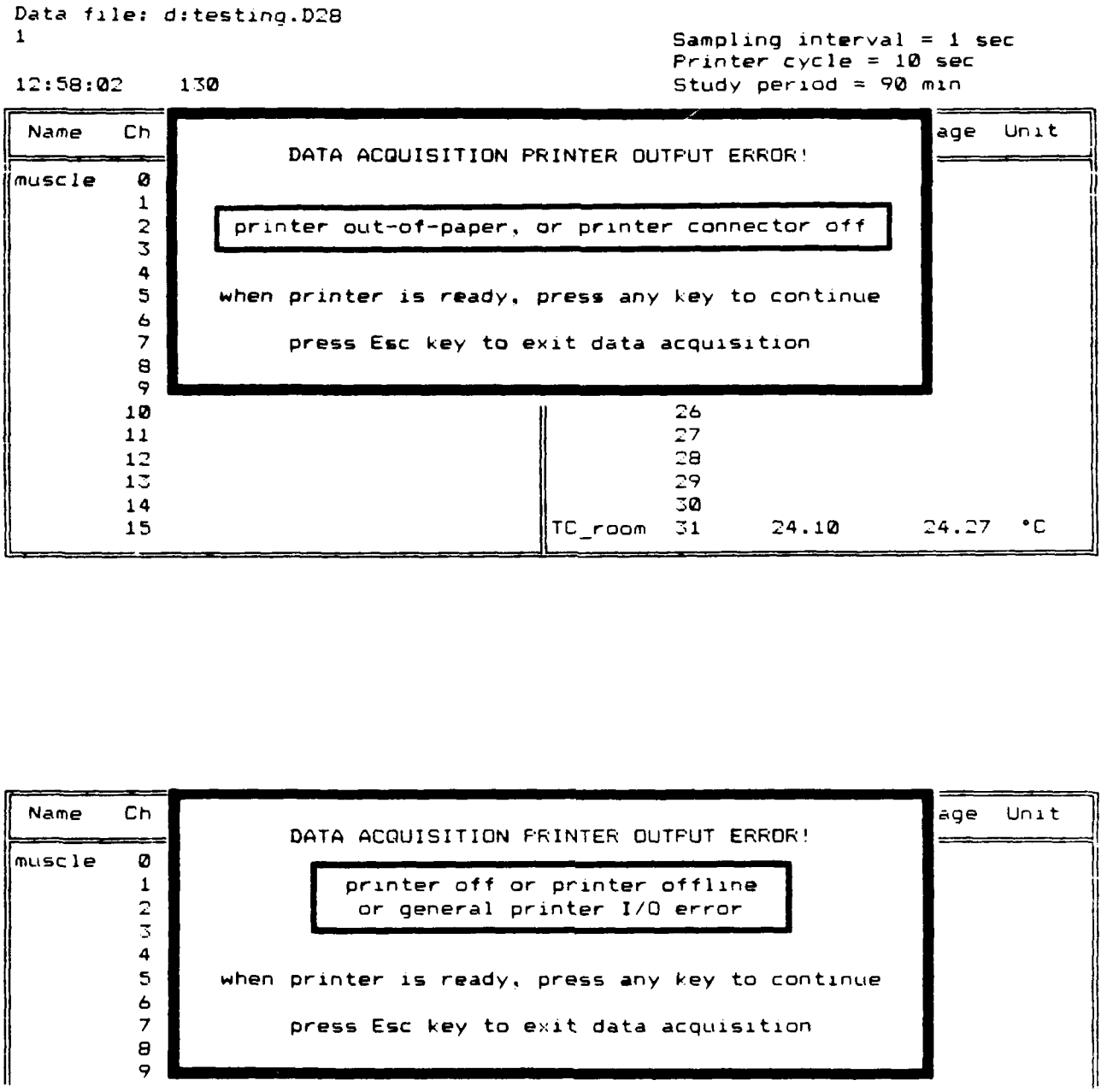


Figure 69 Printer error message windows

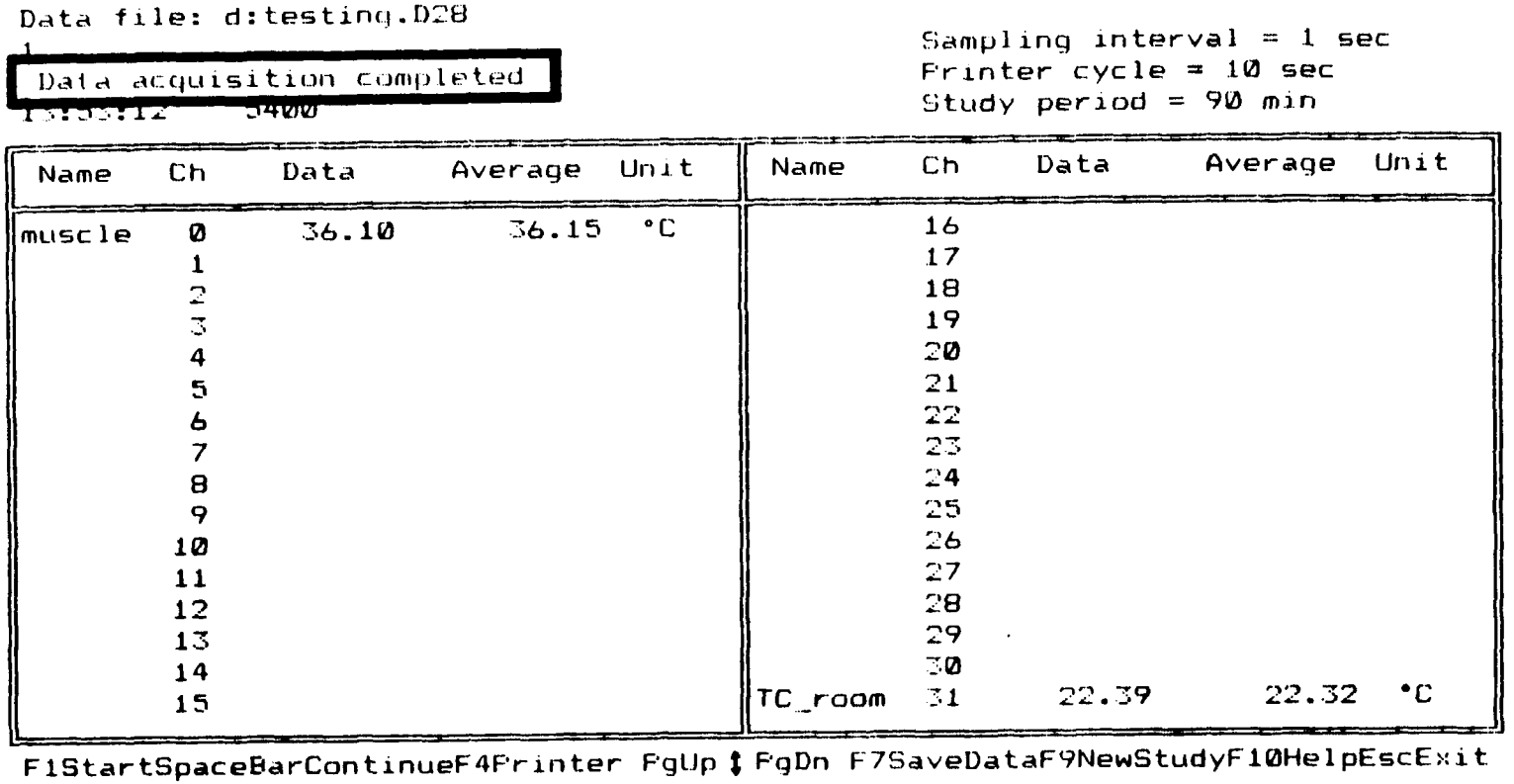


Figure 70 Data acquisition completed message

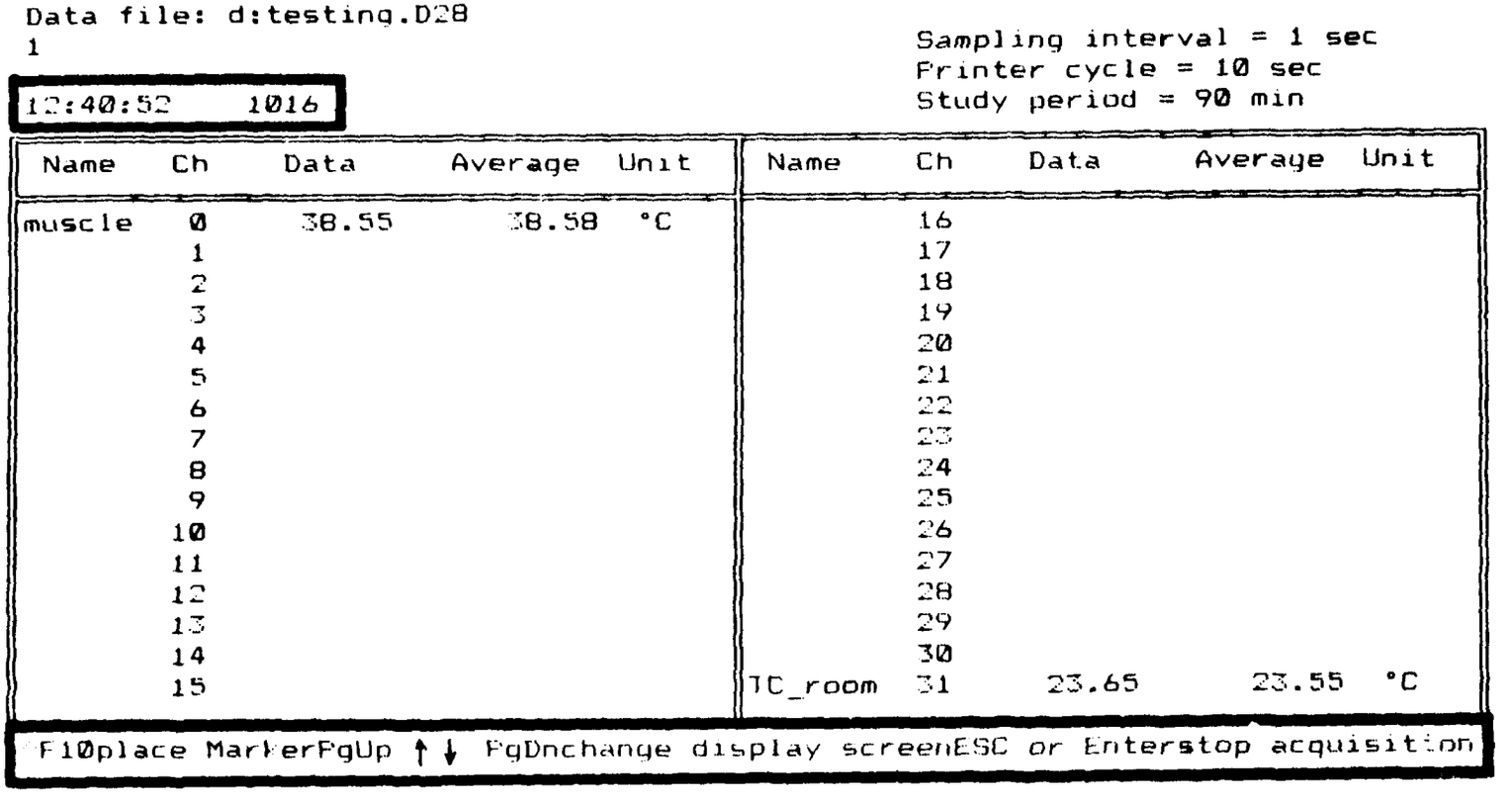


Figure 71 Data Acquisition screen with data

In Figure 71, the F-key row also shows the available functions during data acquisition. The SoS will automatically coordinate the event marker placement and display screen change with the periodic monitor screen data updates. No data will be lost.

Place An Event Marker ^F10–place Marker. An event marker and time are inserted into the data file on the ramdisk and printed on the printer output. This action allows the operator to insert into the data record an indicator signifying a specific or unforseen event that has occurred during a protocol study. The time is also recorded along with the marker. There is no limit to the number of markers that may be inserted into the data file. The marker placed into the data file is "└═══>". If the printer is enabled, a similar marker will be printed. The event marker can be seen in both the sample ramdisk data (Figure 67) and the sample printer output (Figure 68).

^F10 indicates holding down the control (Ctrl) key and pressing the F10 function key simultaneously. This simultaneous key press is purposely programmed to avoid accidental markers. The SoS also beeps three times to notify the operator of the event marker placement.

Change Display Screen PgUp/↑ and PgDn/↓. There are a total of four data display screens and four data plotting screens. These screens can be viewed sequentially by paging up or down. Changing display screen does not affect the normal screen data update. The plotting screen(s) not assigned for plotting, will not be available for viewing during data acquisition.

Temporarily Suspend Data Acquisition ESC/Enter-stop acquisition. This action temporarily suspends the data acquisition. Figure 72 shows the data acquisition suspended message in the Prompt screen. The SoS is returned to the initial Data Acquisition stage. All of the Data Acquisition F-key options are available (see F-key row display of Figure 72). At this stage, acquisition may be resumed by pressing the F1 key or the space bar. Alternatively, data acquisition may be terminated by pressing the ESC or F9 key. The printer may be enabled/disabled, or a different printer number selected. Also, a copy of the data collected up to this point may be saved before resuming acquisition.

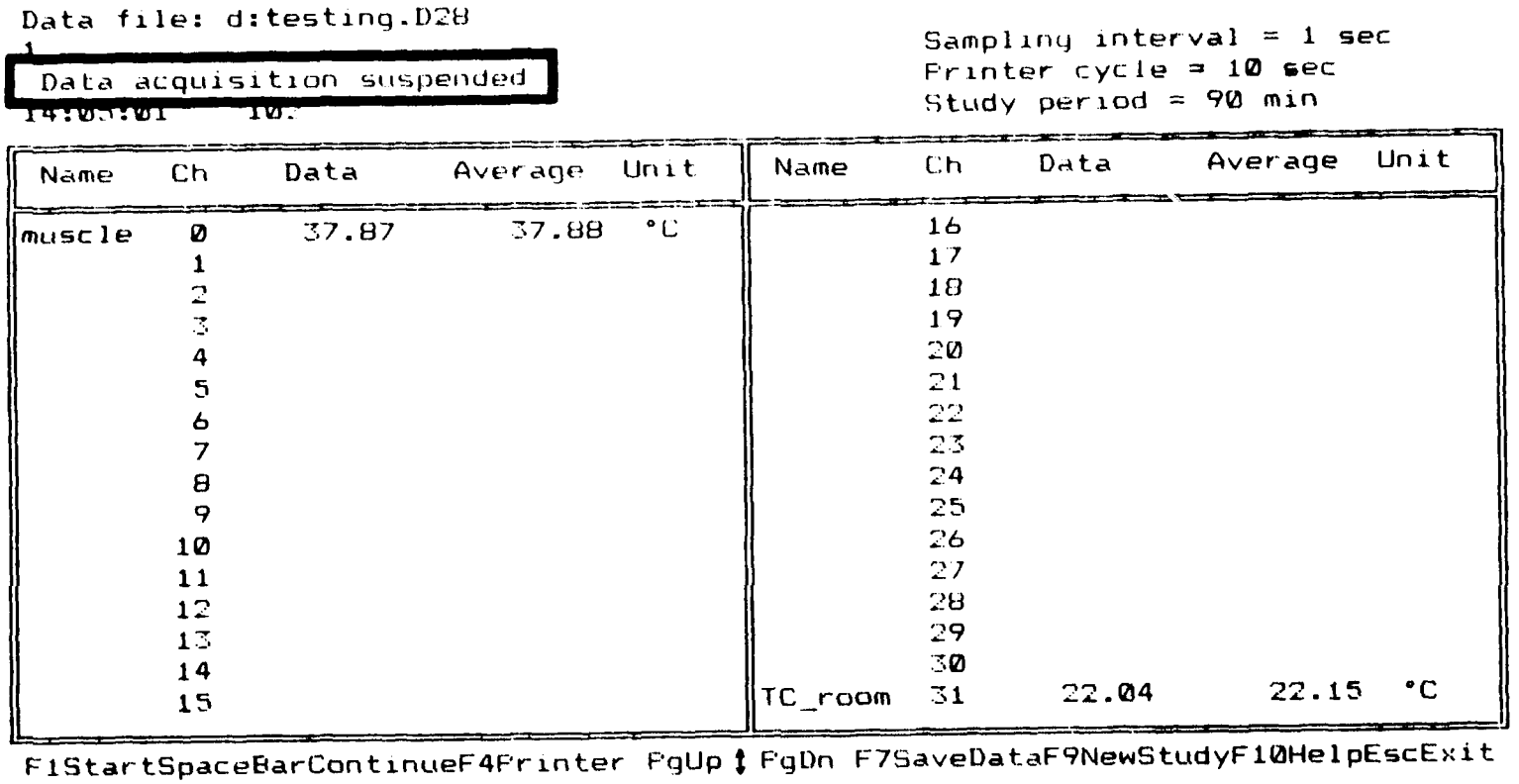


Figure 72 Data acqusition temporarily suspended message

SYSTEM LIMITATIONS

Investigators should be familiar with the limitations of the AC128, in order to avoid expectations that are beyond the capability of the system. The system limitations may be imposed by the hardware or software. In most cases, a combination of both hardware and software factors are involved. The AC128 has been designed to be a general purpose data acquisition system. The advantage of application versatility also carries concomitant disadvantages in other areas, however. In the data collection area, the penalty is data resolution, most acutely in the TM temperature. In the data display area, the penalty is a rather rudimentary graphics capability.

DATA RESOLUTION

The data resolution penalty is most severe for the thermistor data. The TM boards are most susceptible to electrical noise, and do not have onboard hardware filtering capability. The following table is constructed from voltages and resistances measured directly on board the TM board. The temperature data are measured using a quartz thermometer, with calibration traceable to NBS standard. Between each measurement, the voltage difference is approximately 1mV, and the temperature difference is approximately 0.1°C. Hence, to distinguish a 0.1°C difference in TM temperature, the AC128 must have a resolution of 1mV.

TM voltage (mV)	TM resistance (Ω)	temperature (°C)
3611.0	1351.5	37.11
3612.0	1345.1	37.23
3613.4	1335.8	37.40

On the A/D board, the analog-to-digital converter has a 12-bit resolution. The voltage input range for the TM board is from -5V to +5V (±5V), or a total of 10 volts. Therefore,

$$\frac{10 \text{ volt}}{2^{12}} = \frac{10 \text{ volt}}{4096} = 2.44 \text{ mV}$$

The hardware resolution on the A/D board is 2.44 mV. Hence, the hardware resolution is not enough to distinguish a 0.1°C difference in TM temperature. The best resolution is

approximately 0.2°C. With extensive averaging and filtering, incorporated in the software, the average temperature over, for instance, a two minute period may be accurate within 0.1°C (with 1 second sampling rate). However, it would be inappropriate for an investigator or operator to expect a 0.1°C increase/decrease to show on the display monitor instantaneously.

Since the AC128 SoS must also attend to other channels, it simply cannot devote enough time to smooth out each TM temperature input. The investigators should expect some fluctuations in the data displayed on the PC monitor from sampling period to sampling period.

The GP board inputs all use hard potted modules which are less susceptible to electrical noise. All of the GP board modules also have hardware filters incorporated into the circuitry. Hardware filters are faster and more effective than the software filters written in the SoS. For example, the TC measured temperatures are much more stable (less fluctuation) than the TM measured temperatures.

DATA PLOTTING CAPABILITY

Data plotting was included as a visual aid to show the time course change in the data from the selected channel. The data plotting capability is limited partially by the available display memory and partially by QuickBASIC. Currently the data are plotted in the PC display mode 0, essentially a text mode. The plotting quality can benefit greatly if the data can be plotted in graphic mode. QuickBASIC (version 4.0) does not allow different display pages to be assigned to different modes. Since the data display pages are in text mode, this dictates that the plotting pages must also be in text mode. If all display pages are placed in graphic mode then only a maximum of <u>two</u> display pages are available, since the graphic mode display memory demands are much higher. The AC128 requires a minimum of four display pages (4 setup pages). Hence, even before considering any plotting, a graphic mode AC128 would exceed the total display memory resource of the PC system. The graphic mode option cannot be implemented.

Plotting in text mode severely limits the plotting capability and the resolution. To maximize the resolution, only one plot is allowed on each plotting screen. Take the data in Figure 42 as an example. Even by limiting the ordinate to a small range, between 36.0 and 39.6°C, only a 0.2°C y-axis resolution was achieved. Hence, using this plot as an aid to distinguish a 0.1°C change in temperature is inappropriate. The y-axis and time axis resolutions are shown on the left-hand side of the plotting screen.

APPENDIX

AC128 OPERATING SYSTEM SOFTWARE LISTING

The entire software program code is included in this appendix. The program adheres to a modular design. The modules and submodules flow charts are shown in Figures 14 through 19. The software listings are organized into three groups. Group one consists of three of the four main modules: Initial Setup, Main Selection, and Data Acquisition. Group two contains a single module, the Channel Select module, which is written as a subprogram (in QuickBASIC definition). Group three contains the frequently used utility subprograms and Assembly language subroutines.

Logically, the Data Acquisition module is ideally suited to be programed as another subprogram. However, the algorithm in the Data Acquisition module contains the ON TIMER statement which the QuickBASIC restricts usage only in main programs (disallowed in subprograms). The ON TIMER statement drives the input channel sampling procedure, hence cannot be replaced.

In group three, the utility subprograms are mainly interface routines that allow the operator to enter either a name <GetLabel> <GetName>, or a number <GetInteger> <GetNumber>. <ClearLine> is used to clear a line or a rectangle area on the monitor screen. Two other small subprograms <WindowOpen> and <WindowClose> are used with Assembly language subroutines <ScrSave> and <ScrRest>, respectively, to open a dialog box on the screen and to restore the original screen content when done. The Assembly subroutine <PrStat> checks the printer status, and <Acq> commands the A/D board and retrieves the input data.

The entire program contains extensive comments to aid in the understanding of the algorithms. Algorithms at the module and submodule level have also been described in detail in the SOFTWARE section.

```
' ##########################################################################
' ################  AC128 - 128 channel data acquisition system  ################
' ##########################################################################
DECLARE SUB ChannelDisplaySetup128 () '          Data channel setup subprogram
DECLARE SUB WindowOpen (firstline AS INTEGER, lastline AS INTEGER) ' Subprogram
to open a message window
DECLARE SUB WindowClose (firstline AS INTEGER, lastline AS INTEGER) ' Subprogram
to close a message window
DECLARE SUB ClearLine (firstline AS INTEGER, firstcol AS INTEGER, lastline AS
INTEGER, lastcol AS INTEGER, bcolor AS INTEGER) ' Subprogram to clear a line or a
rectangle
DECLARE SUB GetInteger (number AS SINGLE, ESC AS INTEGER, count AS INTEGER,
row AS INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) '
Subprogram to input a printer number from keyboard
DECLARE SUB GetLabel (label$, ESC AS INTEGER, count AS INTEGER, row AS
INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) ' Subprogram to
input a label from keyboard
DECLARE SUB GetName (filename$, ESC AS INTEGER, count AS INTEGER, row AS
INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) ' Subprogram to
input a filename from keyboard

'          parameters passed to subprograms WindowOpen/WindowClose/ClearLine
' firstline    first line of screen to be saved/restored/cleared
' lastline     last line of screen to be saved/restored/cleared

'         parameters passed to subprogram ClearLine
' firstcol    first column to be cleared
' lastcol     last column to be cleared

'           parameters passed to subprograms GetName/GetInteger
' datafile$     acquisition data filename
' setupfile$    screen setup filename
' filename$     path/filename
' label$        label name
' ID$           study run ID remark
' version$      Data Acquisition program title and version
' number         number entered in subprogram
' count         no. of characters in filename
```

```
' fcolor        display foreground color in subprogram
' bcolor        display background color in subprogram

REM $DYNAMIC

COMMON /A/ row AS INTEGER, rowt AS INTEGER, col AS INTEGER, colt AS INTEGER,
coltt AS INTEGER, ch AS INTEGER, cht AS INTEGER, chn AS INTEGER, che AS
INTEGER
COMMON /B/ col1 AS INTEGER, col2 AS INTEGER, col3 AS INTEGER, col4 AS
INTEGER, col5 AS INTEGER, col6 AS INTEGER, col7 AS INTEGER, col8 AS INTEGER,
col9 AS INTEGER, col10 AS INTEGER
COMMON /C/ flagS() AS INTEGER, flagF() AS INTEGER
COMMON /D/ plot() AS SINGLE
COMMON /E/ NC AS INTEGER, ESC AS INTEGER, VS AS INTEGER
COMMON /F/ gain() AS INTEGER, res() AS SINGLE, poly() AS SINGLE
COMMON /G/ samp AS SINGLE, sout AS SINGLE, sper AS SINGLE
COMMON /H/ name$(), volt$(), unit$(), setupfile$
COMMON /I/ ScrStore() AS INTEGER

' row, rowt, rowx              current, temporary row position
' col, colt, colx, coltt      current, temporary column position
' ch, cht                     current, temporary channel number
' chn, che                    total number of availabel/enabled A/D channels
' col1, col2, col3, col4, col5     column 1, 2, 3, 4, 5 position indicator
' col6, col7,,col8, col9, col10    column 6, 7, 8, 9, 10 position indicator

' NC     new column/new screen flag
' ESC    Esc key pressed indicator (1=Esc pressed, 0=not pressed)
' VS     current visual screen: 0=screen 0, 1=screen 1, etc.

DIM flagS(127) AS INTEGER '    select flag
DIM flagF(127) AS INTEGER '    conversion factor flag
DIM gain(127) AS INTEGER 'analg channel select & gain code for subroutine Acq
DIM res(127) AS SINGLE '       data resolution
DIM poly(127, 5) AS SINGLE '   conversion factor array
DIM plot(3, 2) AS SINGLE '             data plot window max & min values
DIM fil1(31), fil2(31) AS SINGLE '    thermistor filtering terms (only 32 TMs)
DIM name$(127), volt$(127), unit$(127) ' channel name, voltage level, unit name
```

```
DIM printsum(127) AS SINGLE '              screen/printer output sum data
DIM coded(127) AS INTEGER ' command code for sub-channel no. (digital output byte)
DIM ScrStore(1999) AS INTEGER '   screen save/restore storage array

' samp, sout, sper      sampling inteval, printer output cycle, study period
'         ssout, ssper, nsamp, nsout must used SINGLE to avoid overflow error
DIM ssout AS SINGLE '   =sout/samp, converts printer output cycle to integer multiple of
sampling interval
DIM ssper AS SINGLE '   =sper*60/samp, converts study period to integer multiple of
sampling inteval
DIM nsamp AS SINGLE '   dummy sample count
DIM nsout AS SINGLE '   dummy printer output count

DIM number AS SINGLE '      number entered in subprogram
DIM DACQ AS INTEGER '       data acquisition flag
DIM flagVS AS SINGLE '      last screen with information to look at (during DataAcq)
DIM rad AS INTEGER '        returned A/D data (from assembly subroutine Acq)
DIM sum AS SINGLE '         temporary sum for A/D data
DIM disp AS SINGLE '        final display data
DIM ad AS SINGLE '          unfiltered averaged A/D data
DIM adn AS INTEGER '        no. of A/D readings to average into 1 A/D data
DIM ohm AS SINGLE '         resistance of reference thermistor
DIM TMvolt AS SINGLE '      Thermistor ground loop adjusted excitation voltage
DIM minusSense AS SINGLE ' Thermistor bridge -sense voltage

'               assembly routine PrStat parameters
DIM PN AS INTEGER '   printer number (1, 2, or 3)
DIM PS AS INTEGER '   printer status byte: bit 3 = printer I/O error

DIM m AS INTEGER, mm AS INTEGER, x AS INTEGER '   dummy variables
DIM n AS INTEGER, nn AS INTEGER, y AS INTEGER
DIM vv AS SINGLE
```

```
version$ = " Biophysics Data Acquisition System ac128.20   900807   Stephen KW. Chang
"

GOSUB NewSetup '                          Initialize screens & data arrays
LOCATE 1, 4: COLOR 14, 3: PRINT version$
LOCATE 4, 1 '                             Help screen reminder
    COLOR 28, 7: PRINT " **** ";
    COLOR 12: PRINT "HELP screens (F10 key) are available ";
    COLOR 28: PRINT "*****"


NextMove:
    GOSUB WhichFile '    New setup file or load old setup file ?
'                        can't separate DataAcq as a SUBprogram
    IF DACQ = 1 THEN '          because of the ON TIMER trap
        GOSUB DataAcq '    data acquisition subroutine (go acquire data)
      ELSE ChannelDisplaySetup128 '   screen setup subprogram
    END IF

    GOTO NextMove
```

```
' ########################## Subroutines start here ############################

' ***********************************************************************************************
' ******************************** Initialize screens & data arrays *****************************
' ***********************************************************************************************
NewSetup:
    chn = 127 '   128 (0 - 127) channels
    che = 0 '     total no. of enabled channels = 0
    nplot = 0 '   number of data plotting channel
    VS = 0 '      set initial visual screen = 0
    flagVS = 7 '  total of 8 visual screens
    ESC = 0
    PN = -1 '     default printer disabled

    col1 = 2: col2 = 9: col3 = 13: col4 = 20: col5 = 33 '      column
    col6 = 41: col7 = 49: col8 = 53: col9 = 60: col10 = 73 '   positions
    col = col2
    row = 8
    ch = 0
    samp = 0: sout = 0: sper = 0 '   sampling time/cycle
    FOR n = 0 TO chn '               data arrays
        name$(n) = "none    "
        volt$(n) = "±       "
        unit$(n) = "none    "
        res(n) = 0
    FOR nn = 0 TO 5: poly(n, nn) = 0: NEXT nn
        flagS(n) = 0
        flagF(n) = 0
    NEXT
    FOR n = 0 TO 31: fil1(n) = 3397: fil2(n) = 3397: NEXT '  3397 ^] TM ≈ 20°C
    FOR n = 0 TO 3: plot(n, 0) = 0: plot(n, 1) = 0: plot(n, 2) = -1: NEXT

' gain() holds gain code in bits 5 & 6 and
' analog input channel select in bits 0 to 4
'                                        ch. 0 - 95 on six 5B02 boards
    FOR n = 0 TO 63 '       A/D        ┌──── channel  0 - 15, 1st 5B02 board
        gain(n) = 15 '     input     1st      channel 16 - 31, 2nd 5B02 board
        coded(n) = n '     channel   STA-20   channel 32 - 47, 3rd 5B02 board
```

```
NEXT '                  #15        └──── channel 48 - 63, 4th 5B02 board
FOR n = 64 TO 95
    gain(n) = 14 '       A/D     2nd ──  channel 64 - 79, 5th 5B02 board
    coded(n) = n - 64 '  #14    STA-20 ─  channel 80 - 95, 6th 5B02 board
NEXT
FOR n = 96 TO 103 '        channel 96 - 103, 1st thermistor EXP-RES board
    gain(n) = 1 '                        map to A/D input channel #1
    coded(n) = n - 96
NEXT
FOR n = 104 TO 111 '       channel 104 - 111, 2nd thermistor EXP-RES board
    gain(n) = 2 '                        map to A/D input channel #2
    coded(n) = n - 104
NEXT
FOR n = 112 TO 119 '       channel 112 - 119, 3rd thermistor EXP-RES board
    gain(n) = 3 '                        map to A/D input channel #3
    coded(n) = n - 112
NEXT
FOR n = 120 TO 127 '       channel 120 - 127, 4th thermistor EXP-RES board
    gain(n) = 4 '                        map to A/D input channel #4
    coded(n) = n - 120
NEXT

FOR mm = 0 TO 7
    SCREEN , , mm, VS
    IF mm < 4 THEN
        GOSUB DataScreen '    draw data screen
      ELSE GOSUB PlotScreen '    draw plotting screen
    END IF
    GOSUB TimePrompt '          time/cycle
NEXT
SCREEN , , VS, VS
RETURN

' ************************************************************************************************
' ************************************ Draw data screen ************************************
' ************************************************************************************************
DataScreen:
  COLOR 1, 7: CLS '                                        clear screen
```

```
   LOCATE 5, 1: PRINT
"╔══════════════════════════════════════╦══════════════════════════════════════
═══╗"
   LOCATE 6, 1: PRINT "║ Name   Ch  Input    Factor     Unit  ║ Name    Ch  Input
Factor      Unit  ║"
   LOCATE 7, 1: PRINT
"╠══════════════════════════════════════╬══════════════════════════════════════
═══╣"
   FOR n = 8 TO 23
       LOCATE n, 1: PRINT "║                                    ║
        ║"
   NEXT n
   LOCATE 24, 1: PRINT
"╚══════════════════════════════════════╩══════════════════════════════════════
═══╝";


   COLOR 8
   nn = 32 * mm '       display channel no. (mm is screen no.)
   FOR n = nn TO nn + 15
       LOCATE row + n - nn, col2: PRINT USING "###"; n: NEXT
   FOR n = nn + 16 TO nn + 31
       LOCATE row + n - nn - 16, col7: PRINT USING "###"; n: NEXT
   ClearLine 25, 1, 25, 80, 3 '   pave F-key row
RETURN


' ***************************************************************************************************
' *************************************  Draw plotting screen  *************************************
' ***************************************************************************************************
PlotScreen:
   COLOR 0, 7: CLS
   LOCATE 5, 17: PRINT
"┬┬────┬────┬────┬────┬────┬────┬────┬────┬────┬────┬────┬────┬┬";
   FOR n = 6 TO 22
       LOCATE n, 17: PRINT "┼                                                              ┼";
   NEXT n
   LOCATE 23, 17: PRINT
```

```
"++---+---+---+---+---+---+---+---+---+---+---++";
    ClearLine 5, 1, 23, 16, 3 '    plot window border
    ClearLine 24, 1, 24, 80, 1 '   time axis
    ClearLine 25, 1, 25, 80, 3 '   F-key row

    IF plot(mm - 4, 2) > -1 THEN '        plot screen scale (mm = screen no.)
        LOCATE 6, 1: COLOR 14: PRINT "ch."; plot(mm - 4, 2)
        LOCATE 7, 1: COLOR 0: PRINT name$(plot(mm - 4, 2))
        LOCATE 12, 2: COLOR 14: PRINT "unit"
        LOCATE 13, 1: COLOR 10: PRINT unit$(plot(mm - 4, 2))
        vv = (plot(mm - 4, 1) - plot(mm - 4, 0)) / 18
        LOCATE 15, 1: COLOR 14: PRINT "resolution"
        LOCATE 16, 1: COLOR 10: PRINT USING "######.##"; vv
        FOR n = 5 TO 23 STEP 3
            LOCATE n, 8: PRINT USING "######.##"; plot(mm - 4, 0) + vv * (23 - n)
        NEXT

        COLOR 15, 1
        LOCATE 23, 1: PRINT "TIME" '              time axis
        LOCATE 24, 1: PRINT "res="; : PRINT USING "#.#"; sper / 60;
                PRINT "  minute  0";
        IF sper < 12 THEN temp$ = "##.#" ELSE temp$ = "####"
        FOR n = 28 TO 80 STEP 10
            LOCATE 24, n - 2: PRINT USING temp$; sper / 60 * (n - 18);
        NEXT
    END IF
RETURN

' ************************************************************************************************************
' ********************************** sampling time/cycle display ***********************************
' ************************************************************************************************************

TimePrompt:
    COLOR 11: ClearLine 2, 49, 4, 79, 0
    LOCATE 2, 49: PRINT " Sampling interval ="; samp; "sec "
    LOCATE 3, 49: PRINT " Printer cycle ="; sout; "sec "
    LOCATE 4, 49: PRINT " Study period ="; sper; "min "
    LOCATE 2, 1: COLOR 6, 7: PRINT mm + 1 '     show page no.
RETURN
```

```
' ************************************************************************************************
' ********************************* Enter new file or Load old file ******************************
' ************************************************************************************************
WhichFile:
    GOSUB WhichFileKey

    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(27) '            Esc - terminate program
            GOSUB ProgramExit

        CASE CHR$(0) + CHR$(72), CHR$(0) + CHR$(73) '  ↑(UpArrow), PgUp(PageUp)
            GOSUB PageUp

        CASE CHR$(0) + CHR$(80), CHR$(0) + CHR$(81) '↓(DownArr), PgDn(PageDown)
            GOSUB PageDown

        CASE CHR$(0) + CHR$(59) '    F1 - Load old setup
            GOSUB LoadSetup

        CASE CHR$(0) + CHR$(60) '    F2 - New setup
            GOSUB NewSetup
            GOTO WhichFile1

        CASE CHR$(0) + CHR$(62) '    F4 - Modify screen
            GOTO WhichFile1

        CASE CHR$(0) + CHR$(63) '    F5 - Save screen setup
            GOSUB SaveSetup

        CASE CHR$(0) + CHR$(65) '    F7 - Save collected data
            GOSUB SaveData

        CASE CHR$(0) + CHR$(66) '    F8 - DOS Shell
            GOSUB DosShell
```

```
        CASE CHR$(0) + CHR$(67) '    F9 - data Acquisition
            DACQ = 1 '           set data acquisition flag = 1
            GOTO WhichFile1

        CASE CHR$(0) + CHR$(68) '    F10 - Main HELP Screen
            GOSUB MainHelp

        CASE ELSE

    END SELECT

'                   from LoadSetup, SaveSetup, SaveData, MainHelp &
    GOTO WhichFile '    No ProgramExit - stay in WhichFile

WhichFile1: '    from NewSetup & Modify - goto ChannelDisplaySetup (DISP128)
RETURN '     from Acquire - goto Data Acquisition

' ************************************* WhichFile F-Key ***************************************
WhichFileKey:
    LOCATE 25, 2
    COLOR 12, 0: PRINT "F1"; : COLOR 10, 3: PRINT "Load";
    COLOR 12, 0: PRINT "F2"; : COLOR 5, 3: PRINT "New";
    COLOR 12, 0: PRINT "F4"; : COLOR 12, 3: PRINT "Modify";
    COLOR 12, 0: PRINT "F5"; : COLOR 11, 3: PRINT "Save setup";
    COLOR 12, 0: PRINT "F7"; : COLOR 9, 3: PRINT "Save data ",
    COLOR 12, 0: PRINT CHR$(18); : COLOR , 3: PRINT " ";
    COLOR 12, 0: PRINT "F8"; : COLOR 1, 3: PRINT "DOS";
    COLOR 12, 0: PRINT "F9"; : COLOR 14, 3: PRINT "Acquire data";
    COLOR 12, 0: PRINT "F10"; : COLOR 15, 3: PRINT "HELP";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "Quit ";
RETURN
```

```
' ************************************************************************************************
' ********************************** filename input window frame **********************************
' ************************************************************************************************
FilenameWindow:
    COLOR 15, 7
    LOCATE 8, 2: PRINT
"╔═══════════════════════════════════════════════════════════════════════════
╗"
    FOR m = 9 TO 23
        LOCATE m, 2: PRINT "‖
        ‖"
    NEXT
    LOCATE 24, 2: PRINT
"╚═══════════════════════════════════════════════════════════════════════════
╝";
RETURN

' ************************************************************************************************
' ******** WAIT!  message window for loading, saving, and restoring setup screen ********
' ************************************************************************************************
WaitWindow:
    WindowOpen 1, 7
    ClearLine 25, 1, 25, 80, 3
    COLOR , 7
    LOCATE 1, 23: PRINT "╔═══════════════════════════════════════╗"
    FOR m = 2 TO 6
        LOCATE m, 23: PRINT "‖                                       ‖"
    NEXT
    LOCATE 7, 23: PRINT "╚═══════════════════════════════════════╝"
RETURN
```

```
' ********************************************************************************************************
' ********** Message window for printer select, printer check, saving collect data ***********
' ********************************************************************************************************
PrinterWindow:
    WindowOpen 5, 25
    ClearLine 25, 1, 25, 80, 3
    COLOR , 7
    LOCATE 5, 13: PRINT
"╔═══════════════════════════════════════════════════════════╗"
    FOR n = 6 TO 16
        LOCATE n, 13: PRINT "‖                                                           ‖"
    NEXT
    LOCATE 17, 13: PRINT
"╚═══════════════════════════════════════════════════════════╝"
RETURN

' ********************************************************************************************************
' ************************** Clear Prompt screen and show setupfile$ ************************
' ********************************************************************************************************
ClearPrompt: '                  used by LoadSetup, SaveSetup, SaveData
    ClearLine 1, 1, 1, 80, 7 '        DataAcq, DataAcqEnd, ProgramExit1
    ClearLine 2, 1, 4, 48, 7
    LOCATE 2, 1: COLOR 2, 1: PRINT mm + 1 '        show page no.
    LOCATE 1, 2: COLOR 15, 4: PRINT " Setup file: "; setupfile$; " "
RETURN
```

```
' ************************************************************************************************************
' ********************************** Program termination point ***********************************************
' ************************************************************************************************************
ProgramExit: '   This is the only place to terminate program and exit
    WindowOpen 1, 4
    ClearLine 1, 1, 4, 80, 7
    ClearLine 25, 1, 25, 80, 3
    COLOR 12, 3: LOCATE 1, 2: PRINT " QUIT!  Are you sure??? "
    COLOR 1: LOCATE 2, 2: PRINT " Have you stored the setup screen? "
    LOCATE 3, 2: PRINT " Have you saved the collected data? "
    COLOR 14: LOCATE 4, 2, 1, 0, 8: PRINT " Confirm quitting (y/n) ";
ProgramExit1:
    k$ = INKEY$
    IF k$ = "" GOTO ProgramExit1
    IF k$ = "y" OR k$ = "Y" THEN '   quit program
        CLS : END
      ELSE '                 not ready to quit
        WindowClose 1, 4
        LOCATE , , 0 '    turn off blinking cursor
        RETURN
    END IF

' ************************************************************************************************************
' ************************************** Go to DOS Shell *****************************************************
' ************************************************************************************************************
DosShell:
    WindowOpen 1, 25
    COLOR 14, 3: CLS
    LOCATE 1, 2
    COLOR 30, 4: PRINT " *****  ";
    COLOR 14:  PRINT " enter 'EXIT' to return to Data Acquisition program  ";
    COLOR 30: PRINT " ***** "
    SHELL
    WindowClose 1, 25
RETURN
```

```
' ************************************************************************************************
' ****************************** Load A/D channel setup information ******************************
' ************************************************************************************************
LoadSetup:
    mm = VS: GOSUB ClearPrompt
    WindowOpen 8, 24 '   open get filename window
    GOSUB FilenameWindow

    LOCATE 9, 32: COLOR 10, 3: PRINT " Load setup file ": COLOR 1, 7
    LOCATE 10, 3: PRINT "Enter setup disk\directory\filename, e.g.
c:my\setup\filename.ext"
    LOCATE 11, 3: PRINT "(64 characters maximum, including ': \'). Only the 8 characters
filename"
    LOCATE 12, 3: PRINT "need to be entered. Directory path and file extension (.ext)
may be"
LOCATE 13, 3: PRINT "omitted. If omitted, the default system disk drive and directory
path will"
    LOCATE 14, 3: PRINT "be assumed. The standard setup file extension is "; : COLOR
15, 2: PRINT "S28"; : COLOR 1, 7: PRINT ", although any other"
    LOCATE 15, 3: PRINT "extension may be used. If no extension is entered, then an ";
: COLOR 15, 2: PRINT "S28"; : COLOR 1, 7: PRINT " will be"
    LOCATE 16, 3: PRINT "appended automatically.": COLOR 10
    LOCATE 17, 3: PRINT " ******* "; : COLOR 1: PRINT "A setup file "; : COLOR 10:
PRINT "MUST ALREADY EXIST "; : COLOR 1: PRINT "and the setup information "; :
COLOR 10: PRINT "********"
    LOCATE 18, 3: PRINT "******** "; : COLOR 1: PRINT "stored in the "; : COLOR 10:
PRINT "CORRECT FORMAT "; : COLOR 1: PRINT "in order to be loaded properly. "; :
COLOR 10: PRINT "********": COLOR 1
    LOCATE 20, 3: PRINT "If unsure about a setup filename, try "; : PRINT "the
demonstration setup file "; : COLOR 15, 2: PRINT "demos": COLOR 1, 7
    LOCATE 22, 3: PRINT "Filename: ";

    IF setupfile$ = "" THEN temp$ = "demos" ELSE temp$ = setupfile$
    GetName temp$, ESC, 64, 22, 13, 10, 7 '   count,row,col,fcolor,bcolor
    WindowClose 8, 24 '                       close get filename window
    IF ESC = 1 GOTO LoadSetup3

    ON ERROR GOTO LoadSetupError '   enables load setup error trap
```

```
LoadSetup1:
    nn = 0
    FOR n = 1 TO LEN(temp$)
        IF MID$(temp$, n, 1) = "." THEN nn = 1 ' check if a file extension
    NEXT '                                      was entered, if not then add
    IF nn = 0 THEN temp$ = temp$ + ".S28" '       .S28 automatically

    GOSUB WaitWindow '      open loading setup message window (WindowOpen 1,7)
    LOCATE 4, 30: PRINT "Loading Setup Screens" '   (COLOR 10,7)

    OPEN temp$ FOR INPUT AS #1
    INPUT #1, samp, sout, sper, che
    FOR n = 0 TO chn
        INPUT #1, flagS(n), flagF(n), gain(n), res(n)
        INPUT #1, poly(n, 5), poly(n, 4), poly(n, 3), poly(n, 2), poly(n, 1), poly(n, 0)
        INPUT #1, name$(n), volt$(n), unit$(n)
    NEXT
    FOR n = 0 TO 3: INPUT #1, plot(n, 2), plot(n, 1), plot(n, 0): NEXT
    CLOSE #1

    ON ERROR GOTO 0 '    disables load setup error trap
    setupfile$ = temp$

LoadSetup2: ' gosub point (from DataAcq) to restore the current setup screen
    FOR n = chn TO 0 STEP -1
        SELECT CASE n
            CASE 0 TO 15
                SCREEN , , 0, VS: rowx = 8: colx = col1
            CASE 16 TO 31
                SCREEN , , 0, VS: rowx = -8: colx = col6
            CASE 32 TO 47
                SCREEN , , 1, VS: rowx = -24: colx = col1
            CASE 48 TO 63
                SCREEN , , 1, VS: rowx = -40: colx = col6
            CASE 64 TO 79
                SCREEN , , 2, VS: rowx = -56: colx = col1
            CASE 80 TO 95
                SCREEN , , 2, VS: rowx = -72: colx = col6
```

```
    CASE 96 TO 111
        SCREEN , , 3, VS: rowx = -88: colx = col1
    CASE ELSE
        SCREEN , , 3, VS: rowx = -104: colx = col6
END SELECT

LOCATE rowx + n, colx
IF name$(n) = "none    " THEN '    Device name
    COLOR , 7: PRINT SPC(7);
  ELSE COLOR 0, 3: PRINT name$(n);
END IF '                          there is a space between columns 6 & 7
IF colx = col6 THEN COLOR , 7: PRINT " "; '  only (not columns 1 & 2)

IF flagS(n) >= 2 THEN '    Selected channels
    COLOR 14, 7
  ELSEIF flagS(n) = 1 THEN
    COLOR 15, 7
  ELSE COLOR 8, 7
END IF
PRINT USING "###"; n; : PRINT " ";

IF volt$(n) <> "±      " THEN '              Input voltage/Gain code
    COLOR 14, 4: PRINT volt$(n); : COLOR , 7: PRINT " ";
  ELSE COLOR , 7: PRINT SPC(7);
END IF

COLOR 15, 1 '              Conversion factors
    SELECT CASE flagF(n)
        CASE 8
            PRINT "thermocouple";
        CASE 4
            PRINT " thermistor ";
        CASE 2
            PRINT "   linear   ";
        CASE 1
            PRINT " polynomial ";
        CASE ELSE
            COLOR , 7: PRINT SPC(12);
```

```
            END SELECT
        COLOR , 7: PRINT " ";

        IF unit$(n) = "none    " THEN '     Unit labels
            COLOR , 7: PRINT SPC(7);
          ELSE COLOR 0, 2: PRINT unit$(n);
        END IF
    NEXT n

    IF ESC <> 16 THEN '     ESC=16 => LoadSetup after DataAcq, thus
        FOR mm = 4 TO 7 '               don't erase plot screens
            SCREEN , , mm, VS: GOSUB PlotScreen '  else new plot screens
        NEXT mm
        ESC = 0
    END IF

    WindowClose 1, 7 '   close loading setup message window

    FOR mm = 0 TO 7 '          it's necessary to display time/cycle
        SCREEN , , mm, VS '    after closing message window
        GOSUB TimePrompt '     time/cycle display
        GOSUB ClearPrompt '    show setupfile$
    NEXT
    SCREEN , , VS, VS

LoadSetup3:
RETURN


LoadSetupError:
    CLOSE #1
    WindowClose 1, 7 '   close loading setup message window
    BEEP
    WindowOpen 8, 24 '  (COLOR 10, 7) open loading setup error message window
    GOSUB FilenameWindow

    LOCATE 9, 14: COLOR 28, 3: PRINT " ******** "; : COLOR 10: PRINT "Unable to find
or load setup file "; : COLOR 28: PRINT "******** ": COLOR 1, 7
```

```
    LOCATE 11, 3: PRINT "Is the setup information in the "; : COLOR 10: PRINT
"CORRECT FORMAT"; : COLOR 1: PRINT "?"
    LOCATE 12, 3: PRINT "Check directory\filename path.  Setup file "; : COLOR 10:
PRINT "MUST ALREADT EXIST"; : COLOR 1: PRINT "."
    LOCATE 13, 3: PRINT "If loading from a hard disk, does the directory path exists?"
    LOCATE 14, 3: PRINT "If loading from floppy disk, was the disk drive door closed?
Reinsert"
    LOCATE 15, 3: PRINT "floppy disk and try again.  Disk may be damaged, try backup
disk.  If"
    LOCATE 16, 3: PRINT "available, try another floppy disk drive.  Enter setup
directory\filename"
    LOCATE 17, 3: PRINT "again, e.g. c:my\setup\filename.ext (64 characters maximum,
including"
    LOCATE 18, 3: PRINT "': \').  Only the 8 characters filename is required.  If no
extension is "
    LOCATE 19, 3: PRINT "entered, then an "; : COLOR 15, 2: PRINT "S28"; : COLOR 1,
7: PRINT " will be appended automatically."
    LOCATE 20, 3: PRINT "If unsure about a setup filename, try the demonstration setup
file "; : COLOR 15, 2: PRINT "demos": COLOR 1, 7:
    LOCATE 22, 3: PRINT "Filename: ";

    GetName temp$, ESC, 64, 22, 13, 10, 7 '    count,row,col,fcolor,bcolor
    WindowClose 8, 24 '            close loading setup error message window

    IF ESC = 1 THEN RESUME LoadSetup3 ELSE RESUME LoadSetup1
```

```
' ************************************************************************************************
' ****************************** Save A/D channel setup information ******************************
' ************************************************************************************************
SaveSetup:
    mm = VS: GOSUB ClearPrompt
    WindowOpen 8, 24 '   open get filename window
    GOSUB FilenameWindow

    LOCATE 9, 32: COLOR 11, 3: PRINT " Save setup file ": COLOR 1, 7
    LOCATE 10, 3: PRINT "Enter setup disk\directory\filename, e.g.
c:my\setup\filename.ext"
    LOCATE 11, 3: PRINT "(64 characters maximum, including ': \'). Only the 8 characters
filename"
    LOCATE 12, 3: PRINT "need to be entered. Disk drive, directory path and file
extension (.ext)"
    LOCATE 13, 3: PRINT "may be omitted. If omitted, the default system disk drive and
directory"
    LOCATE 14, 3: PRINT "will be assumed. The standard setup file extension is "; :
COLOR 15, 3: PRINT "S28"; : COLOR 1, 7: PRINT ", although any"
    LOCATE 15, 3: PRINT "other extension may be used. It is recommended that you
stay with the"
    LOCATE 16, 3: PRINT "suggested designation to avoid possible setup file confusion
later on."
    LOCATE 17, 3: PRINT "If no extension is entered, then an "; : COLOR 15, 3: PRINT
"S28"; : COLOR 1, 7: PRINT " will be appended automatically.": COLOR 11
    LOCATE 19, 3: PRINT "   ****** "; : COLOR 1: PRINT "If supplying a disk drive name,
is the disk drive "; : COLOR 11: PRINT "READY"; : COLOR 1: PRINT "? "; : COLOR 11:
PRINT "*******"
    LOCATE 20, 3: PRINT " ****** "; : COLOR 1: PRINT "If supplying a directory path,
directory "; : COLOR 11: PRINT "MUST ALREADY EXIST"; : COLOR 1: PRINT ". "; :
COLOR 11: PRINT "*******": COLOR 1
    LOCATE 22, 3: PRINT "Filename: ";

    temp$ = setupfile$
    GetName temp$, ESC, 64, 22, 13, 11, 7 '  count,row,col,fcolor,bcolor
    WindowClose 8, 24 '                      close get filename window
    IF ESC = 1 GOTO SaveSetup2
```

```
    ON ERROR GOTO SaveSetupError '   enables save setup error trap
SaveSetup1:
    nn = 0
    FOR n = 1 TO LEN(temp$)
        IF MID$(temp$, n, 1) = "." THEN nn = 1
    NEXT '                              check if a file extension was entered,
    IF nn = 0 THEN temp$ = temp$ + ".S28" 'if not then add .S28 automatically

    GOSUB WaitWindow '      open saving setup message window (WindowOpen 1,7)
    LOCATE 4, 30: PRINT "Saving Setup Screens" '   (COLOR 11,7)

    OPEN temp$ FOR OUTPUT AS #1
    PRINT #1, samp; sout; sper; che
    FOR n = 0 TO chn
        PRINT #1, flagS(n); flagF(n); gain(n); res(n);
        PRINT #1, poly(n, 5); poly(n, 4); poly(n, 3); poly(n, 2); poly(n, 1); poly(n, 0);
        PRINT #1, CHR$(34); name$(n); CHR$(34);
        PRINT #1, CHR$(34); volt$(n); CHR$(34);
        PRINT #1, CHR$(34); unit$(n); CHR$(34)
    NEXT
    FOR n = 0 TO 3: PRINT #1, plot(n, 2); plot(n, 1); plot(n, 0): NEXT
    CLOSE #1

    ON ERROR GOTO 0 '   disables save setup error trap
    WindowClose 1, 7 '  close saving setup message window

    setupfile$ = temp$
    FOR mm = 0 TO 7
        SCREEN , , mm, VS
        GOSUB ClearPrompt: LOCATE 2, 14: PRINT " SAVED "
        COLOR , 1
        LOCATE 3, 2: PRINT " REMINDER: to print a copy of this screen "
        LOCATE 4, 2: PRINT "           press Print Screen (PrtSc) key "
    NEXT
    SCREEN , , VS, VS

SaveSetup2:
RETURN
```

```
SaveSetupError:
    WindowClose 1, 7 '   close saving setup message window
    BEEP
    WindowOpen 8, 24 '   (COLOR 11, 7) open saving setup error message window
    GOSUB FilenameWindow

    LOCATE 9, 13: COLOR 28, 3: PRINT " ******** "; : COLOR 11: PRINT "Unable to
store or write setup file "; : COLOR 28: PRINT "******** ": COLOR 1, 7
    LOCATE 11, 3: PRINT "Check disk\directory\filename path."
    LOCATE 12, 3: PRINT "If supplied a disk drive name, was the disk drive "; : COLOR
11: PRINT "READY"; : COLOR 1: PRINT "?"
    LOCATE 13, 3: PRINT "If supplied a directory path, the directory "; : COLOR 11:
PRINT "MUST ALREADY EXIST"; : COLOR 1: PRINT "."
    LOCATE 14, 3: PRINT "If storing to a floppy disk, was the disk drive door closed?
Reinsert"
    LOCATE 15, 3: PRINT "floppy disk and try again.  Disk may be full, use a blank
formatted disk."
    LOCATE 16, 3: PRINT "If available, try another disk drive.  Enter setup
directory\filename again,"
    LOCATE 17, 3: PRINT "e.g. c:my\setup\filename.ext (64 characters maximum, including
': \')."
    LOCATE 18, 3: PRINT "Only the 8 characters filename is required.  The standard file
extension "; : COLOR 15, 3: PRINT "S28"; : COLOR 1, 7
    LOCATE 19, 3: PRINT "is recommended, although any other may be used."
    LOCATE 20, 3: PRINT "If no extension is entered, then an "; : COLOR 15, 3: PRINT
"S28"; : COLOR 1, 7: PRINT " will be appended automatically."
    LOCATE 22, 3: PRINT "Filename: ";

    GetName temp$, ESC, 64, 22, 13, 11, 7 '   count,row,col,fcolor,bcolor
    WindowClose 8, 24 '            close saving setup error message window

    IF ESC = 1 THEN RESUME SaveSetup2 ELSE RESUME SaveSetup1
```

```
' ************************************************************************************************
' ************************************ Save collected data file ************************************
' ************************************************************************************************
SaveData:
    mm = VS: GOSUB ClearPrompt
    WindowOpen 8, 24 '   open get filename window
    GOSUB FilenameWindow

    LOCATE 9, 3: COLOR 9, 3: PRINT "Save collected data file"; : COLOR 1, 7: PRINT "
("; : COLOR 15, 1: PRINT "SOURCE"; : COLOR 1, 7: PRINT ") on a more permanent
storage device, e.g. "
    LOCATE 10, 3: PRINT "hard or floppy disk ("; : COLOR 15, 1: PRINT
"DESTINATION"; : COLOR 1, 7: PRINT "). Need both SOURCE and DESTINATION
names.";
    LOCATE 12, 3: PRINT "Enter "; : COLOR 15, 1: PRINT "SOURCE"; : COLOR 1, 7:
PRINT " data filename first. Data should be on a ramdisk, and can"
    LOCATE 13, 3: PRINT "generally be retrieved by the following filename format
d:filename.ext --> "
    LOCATE 14, 3: PRINT "2 characters disk drive, 8 characters filename, 3 characters
extension "
    LOCATE 15, 3: PRINT "(excluding the '.'), although longer filename path could have
been used"
    LOCATE 16, 3: PRINT "during data collection.  Only the 8 characters filename needs
to be entered."
    LOCATE 17, 3: PRINT "if disk drive and extension are omitted, the default ramdisk
designation "; : COLOR 15, 1: PRINT "d:": COLOR 1, 7
    LOCATE 18, 3: PRINT "and the standard data file extension "; : COLOR 15, 1: PRINT
"D28"; : COLOR 1, 7: PRINT " will be appended automatically.": COLOR 15, 4
    LOCATE 19, 3: PRINT "CAUTION: incorrect/nonexisting data filename may not be
detected early"
    LOCATE 20, 3: PRINT "enough to avoid copying error.  Be sure to use correct data
filename.": COLOR , 1
    LOCATE 21, 3: PRINT "SOURCE"
    LOCATE 22, 3: PRINT "datafile:"; : COLOR , 7: PRINT " ";

    GetName datafile$, ESC, 64, 22, 13, 9, 7 '  count,row,col,fcolor,bcolor
    IF ESC = 1 THEN '     ESC key pressed
        WindowClose 8, 24 '   close get filename window
```

```
        GOTO SaveData1
    END IF

    mm = 0: nn = 0 '  check if disk drive and file extension were entered
    FOR n = 1 TO LEN(datafile$)
        IF MID$(datafile$, n, 1) = ":" THEN mm = n '    disk: entered
        IF MID$(datafile$, n, 1) = "." THEN nn = n '    .ext entered
    NEXT
    IF mm = 0 THEN datafile$ = "d:" + datafile$ '  no disk drive, appended d:
    IF nn = 0 THEN datafile$ = datafile$ + ".D28" '  no .ext, append .D28

    LOCATE 11, 3: COLOR 15, 1: PRINT "SOURCE";
    IF LEN(datafile$) < 61 THEN PRINT ' datafile"; '      datafile + d: + .D28
    COLOR 9, 7: PRINT " "; datafile$ '              would be too long to display

    COLOR 1: ClearLine 12, 3, 23, 78, 7
    LOCATE 13, 3: PRINT "Now enter "; : COLOR 15, 1: PRINT "DESTINATION"; :
COLOR 1, 7: PRINT " disk\directory\filename, e.g. c:my\data\filename.ext"
    LOCATE 14, 3: PRINT "(64 characters maximum, including ': \'). Only the 8 characters
filename"
    LOCATE 15, 3: PRINT "is needed.  If directory path is omitted, the default system disk
drive and"
    LOCATE 16, 3: PRINT "directory will be assumed.  "; : COLOR 9: PRINT "No file
extension is assumed.  "; : COLOR 15, 4
    LOCATE 17, 3: PRINT "CAUTION: incorrect\nonexisting directory or filename, or
not-ready floppy"
    LOCATE 18, 3: PRINT "disk drive may not be detected early enough to avoid copying
error. ": COLOR 1, 7
    LOCATE 19, 3: PRINT "In the Prompt area, a "; : COLOR 9: PRINT "1 File(s) copied
"; : COLOR 1: PRINT "message indicates a successful data ";
    LOCATE 20, 3: PRINT "file save.  Other messages may show copying error, and the
cause of error.": COLOR 15, 1
    LOCATE 21, 3: PRINT "DESTINATION"
    LOCATE 22, 3: PRINT "filename:"; : COLOR , 7: PRINT " ";

    GetName filename$, ESC, 64, 22, 13, 9, 7 '  count,row,col,fcolor,bcolor
    WindowClose 8, 24 '                          close get filename window
    IF ESC = 1 GOTO SaveData1
```

```
    GOSUB PrinterWindow '   open saving data message window (WindowOpen 5,25)
    LOCATE 8, 30: PRINT "Saving Collected data" '    (COLOR 9,7)
    LOCATE 12, 21: PRINT "(watch for messages in the Prompt area)"

    FOR mm = 0 TO 7
        SCREEN , , mm, VS
        GOSUB ClearPrompt
        ClearLine 1, 1, 1, 80, 7
        LOCATE 1, 1 '                          if filename$ has full 64 char. and
        IF LEN(datafile$) + LEN(filename$) < 72 THEN ' datafile$ has 12 char.
            COLOR 15, 1: PRINT datafile$; '        then don't print datafile$
        END IF '                                  else line will be too long
        COLOR 7: PRINT " save to "; : COLOR 15: PRINT filename$;
    NEXT
    SCREEN , , VS, VS
'                       copy message will appear only on current video page
    COLOR 9: ClearLine 2, 1, 4, 48, 7 '  prepare copy status area (COLOR 9,7)
    LOCATE 2, 1: SHELL "copy " + datafile$ + " " + filename$

    WindowClose 5, 25 '   close saving data message window

SaveData1:
RETURN
```

```
' ************************************************************************************************
' ********************************************** MainHelp ****************************************
' ************************************************************************************************
MainHelp:
   WindowOpen 1, 25 '    open HELP screen window
   GOTO MainHelpPage1 '  display Main HELP page 1

MainHelp1:
   k$ = INKEY$

   SELECT CASE k$

      CASE CHR$(0) + CHR$(80), CHR$(0) + CHR$(81) ' ↓(DownArrow), PageDown
         GOTO MainHelpPage2

      CASE CHR$(0) + CHR$(72), CHR$(0) + CHR$(73) '  ↑(UpArrow), PgUp(PageUp)
         GOTO MainHelpPage1

      CASE CHR$(27), CHR$(0) + CHR$(68) '   Esc & F10 - leave HELP screen
         WindowClose 1, 25 '              close HELP screen window
         RETURN

      CASE ELSE
         GOTO MainHelp1

   END SELECT

MainHelpPage1: ' ***************************** Main HELP page 1 ******************************
   COLOR 15, 7
   LOCATE 1, 1: PRINT " "; : COLOR , 0: PRINT "  GENERAL PURPOSE DATA
ACQUISITION PROGRAM (128.20) - 128 channels available  "; : COLOR , 7: PRINT " "
   LOCATE 2, 1: PRINT "  "; : COLOR , 0: PRINT " Each data channel must be setup
before data collection can begin.  Setup   "; : COLOR , 7: PRINT "  "
   LOCATE 3, 1: PRINT "  "; : COLOR , 0: PRINT " procedure involves selecting the
input data type and indicating the output "; : COLOR , 7: PRINT "  "
   LOCATE 4, 1: PRINT "  "; : COLOR , 0: PRINT " data processing for each channel.
Each channel setup is begun by choosing "; : COLOR , 7: PRINT "  "
   LOCATE 5, 1: PRINT "  "; : COLOR , 0: PRINT " the appropriate F-key function, and
```

```
then follow the prompt instructions.  "; : COLOR , 7: PRINT "  "
   LOCATE 6, 1:  PRINT SPC(19); :                COLOR 1: PRINT
"╔═══════════════════════╦═══════════════════════╗"; SPC(20); : COLOR 0
   LOCATE 7, 1:  PRINT "  Each screen is   "; : COLOR 1: PRINT "‖          1
 ‖     2       ‖"; : COLOR 0: PRINT "     There are 4    "
   LOCATE 8, 1:  PRINT "   divided into    "; : COLOR 1: PRINT
"‖═══════════════════════╩═══════════════════════╣"; : COLOR 0: PRINT "   display screens
"
   LOCATE 9, 1:  PRINT "     4 areas       "; : COLOR 1: PRINT "‖                  ‖
          ‖"; : COLOR 0: PRINT "    each contains   "
   LOCATE 10, 1: PRINT "  (not drawn to    "; : COLOR 1: PRINT "‖            3     ‖
 3             ‖"; : COLOR 0: PRINT "     32 channels    "
   LOCATE 11, 1: PRINT "   proportion)     "; : COLOR 1: PRINT
"‖═══════════════════════════════════════╣"; : COLOR 0: PRINT "      of setup
"
   LOCATE 12, 1: PRINT SPC(19); :                COLOR 1: PRINT "‖                  4
          ‖"; : COLOR 0: PRINT "     information    "
   LOCATE 13, 1: PRINT SPC(19); :                COLOR 1: PRINT
"╚═══════════════════════════════════════╝"; SPC(20);
   COLOR 0
   LOCATE 14, 1: PRINT " Area (1) is the PROMPT screen.  System status, acquisition
channel information,"
   LOCATE 15, 1: PRINT "          program warning and error messages are displayed in
this area.        "
   COLOR 1
   LOCATE 16, 1: PRINT " Area (2) is the TIME/CYCLE screen.  Data channel sampling
interval, study      "
   LOCATE 17, 1: PRINT "          period, and printer output cycle informations are
shown.            "
   COLOR 0
   LOCATE 18, 1: PRINT " Area (3) is the channel setup display screen.  This area (each
half) is       "
   LOCATE 19, 1: PRINT "          divided into 5 columns: Device Name, Channel
Number, Input Voltage    "
   LOCATE 20, 1: PRINT "          Range, Conversion Factor, and Unit Label.  The
column currently      "
   LOCATE 21, 1: PRINT "          active has a "; : COLOR 10, 0: PRINT "highlighted
heading"; . COLOR 0, 7: PRINT ", and a "; : COLOR 16, 3: PRINT "blinking label"; :
```

```
COLOR 0, 7: PRINT " indicates       "
    LOCATE 22, 1: PRINT "           the currently active channel.
          "
    COLOR 1
    LOCATE 23, 1: PRINT " Area (4) is the F-key row.  When this row is shown, only the
indicated keys     "
    LOCATE 24, 1: PRINT "           are activated.  Other keys have no effect.
            ";
    COLOR 0, 3
    LOCATE 25, 1: PRINT "  ....MORE....     "; : COLOR 15: PRINT CHR$(25); : COLOR
0: PRINT " or "; : COLOR 15: PRINT "PgDn"; : COLOR 0: PRINT " for next screen     "; :
COLOR 15: PRINT "Esc"; : COLOR 0: PRINT " or "; : COLOR 15: PRINT "F10"; : COLOR
0: PRINT " to leave Help screen  ";
    GOTO MainHelp1

MainHelpPage2: ' ****************************** Main HELP page 2 ******************************
    COLOR 0, 7
    LOCATE 1, 2: PRINT "When the F-key row is shown (always the last row on the
screen), only the       "
    LOCATE 2, 2: PRINT "indicated keys are activated.  F-keys select particular functions.
Arrows     "
    LOCATE 3, 2: PRINT "move between columns and rows.  PgUp and PgDn move
between the 4 setup          "
    LOCATE 4, 2: PRINT "display screens.  The Esc key always exits to the previous step.
Before       "
    LOCATE 5, 2: PRINT "exiting the program, you will be prompted for confirmation and
reminded to      "
    LOCATE 6, 2: PRINT "save everything.  "; : COLOR 15, 0: PRINT "Note:"; : COLOR
0, 7: PRINT " setup information not intentionally saved will be lost."
    LOCATE 7, 2: COLOR 12, 0: PRINT "F1"; : COLOR 10, 7: PRINT " - loads old setup
file.  File must already exist (i.e. saved previously).    "
    LOCATE 8, 2: PRINT "      If unsure about a filename, try the demonstration setup file
"; : COLOR 15, 2: PRINT "demos"; : COLOR 10, 7: PRINT ".          "
    LOCATE 9, 2: COLOR 12, 0: PRINT "F2"; : COLOR 5, 7: PRINT " - Use F2 to enter
setup information into a new (blank) setup screen.          "
    LOCATE 10, 2: COLOR 12, 0: PRINT "F4"; : COLOR 12, 7: PRINT " - modifies current
setup screen.  When there is setup information already    "
    LOCATE 11, 2: PRINT "      on the screen (loaded previously), the setup information of
```

```
each data      "
    LOCATE 12, 2: PRINT "      channel can be modified.  New setup file can be created
by modifying an    "
    LOCATE 13, 2: PRINT "      old setup screen, then saving the updated screen into a
new setup file.    "
    LOCATE 14, 2: COLOR 12, 0: PRINT "F5"; : COLOR 11, 7: PRINT " - stores the
current setup screen into a "; : COLOR 15, 3: PRINT "setup file"; : COLOR 11, 7: PRINT
".  You will be asked for "
    LOCATE 15, 2: PRINT "      a setup filename.  Channel setup information not saved
will be lost.      "
    LOCATE 16, 2: COLOR 12, 0: PRINT "F7"; : COLOR 9, 7: PRINT " - saves collected
data into a "; : COLOR 15, 1: PRINT "data file"; : COLOR 9, 7: PRINT ".  This option is
also available     "
    LOCATE 17, 2: PRINT "      during data acquisition.  Save all the collected data as
soon as possible."
    LOCATE 18, 2: COLOR 12, 0: PRINT "F8"; : COLOR 1, 7:  PRINT " - goes to a DOS
shell.  Enter 'EXIT' to return to Data Acquisition program.  "
    LOCATE 19, 2: COLOR 12, 0: PRINT "F9"; : COLOR 14, 7: PRINT " - starts the data
acquisition procedure.  Start acquire data only when      "
    LOCATE 20, 2: PRINT "      channel setup is completed.  You will be prompted for a
data filename.     "
    LOCATE 21, 2: COLOR 12, 0: PRINT "F10"; : COLOR 15, 7: PRINT " - Different
HELP screens are available.  Each setup column: Device Name,    "
    LOCATE 22, 2: PRINT "       Channel Number, Input Voltage Range, Conversion
Factor, & Unit Label      "
    LOCATE 23, 2: PRINT "      has its own HELP screen.  Whenever stuck, try the F10 -
HELP screen.      "
    LOCATE 24, 7: COLOR 15, 0: PRINT "  "; CHR$(24); : COLOR 7: PRINT " or "; :
COLOR 15: PRINT "PgUp"; : COLOR 7: PRINT " for previous screen     "; : COLOR 15:
PRINT "Esc"; : COLOR 7: PRINT " or "; : COLOR 15: PRINT "F10"; : COLOR 7: PRINT  "
to leave Help screen  "; : COLOR , 7: PRINT SPC(6);
    GOSUB WhichFileKey
    GOTO MainHelp1
```

```
' ************************************************************************************************
' ************************************** Data acquisition ****************************************
' ************************************************************************************************
' Acq(commmand code, digital output byte, rad) assembly data acq. subroutine
'    command code = analog input channel select  OR  channel gain
'    digital output byte = specifies input sub-channel no.,
'                          will be inverted before CALL Acq subroutine
'    rad = returned A/D data (from subroutine Acq)

' PrStat(PN, PS)   assembly subroutine to detect printer status
'    PN = printer number (1, 2, 3)
'    PS = printer status byte: bit 3 = printer I/O error
'                              bit 5 = out-of-paper error

DataAcq:
   IF samp = 0 OR sper = 0 THEN '   no sampling interval or no study period
      FOR mm = 0 TO 7
         SCREEN , , mm, VS
         COLOR 28, 0: LOCATE 2, 49: PRINT " NO sampling interval          "
         COLOR 20: LOCATE 3, 49: PRINT " or, NO study period              "
         COLOR 12: LOCATE 4, 49: PRINT " select F4 and then Timing key "
      NEXT
      BEEP: VS = 0: ch = 0: row = 8: col = col2
      SCREEN , , VS, VS
      GOTO DataAcqEnd1
   END IF
   FOR mm = 0 TO 7 '                        erase Prompt & restore setupfile$,
      SCREEN , , mm, VS: GOSUB ClearPrompt ' in case SaveData
   NEXT '                                      or SaveSetup was done
   SCREEN , , VS, VS

DataAcq1:
   GOSUB DataAcqFilename '              Ask for display data storage filename
   IF ESC = 1 GOTO DataAcqEnd1 '   if Esc key was pressed then go to
'                                 data acquisition exit routine
   ON ERROR GOTO DataAcqFilenameError ' incorrect data filename error trap
   OPEN datafile$ FOR OUTPUT AS #2 ' open data storage file on ramdisk (d:)
   PRINT #2, version$
```

```
    PRINT #2, DATE$; "  "; datafile$; "  "; ID$; '   store date & ID remark
    CLOSE #2
    ON ERROR GOTO 0 '   disable incorrect data filename error trap

    ssout = sout / samp ' convert printer output cycle to multiple of sampling interval
    ssper = sper * 60 / samp ' convert study period to multiple of sampling interval
    nsamp = 0 '                 sample count
    nsout = ssout '             printer output count
    FOR n = 0 TO chn: printsum(n) = 0: NEXT 'initialize printer & average buffer

    GOSUB DataAcqScreen '           data acquisition display screen
    GOSUB DataAcqPrinterSelect3 '   display printer number used

DataAcq2:
    GOSUB DataAcqKey

    k$ = INKEY$

        SELECT CASE k$

            CASE CHR$(27), CHR$(0) + CHR$(67) '    Esc - End data acquisition
                GOTO DataAcqEnd '                  F9 - New (another) study

            CASE CHR$(0) + CHR$(72), CHR$(0) + CHR$(73) ' ↑(UpArrow), PageUp
                GOSUB PageUp

            CASE CHR$(0) + CHR$(80), CHR$(0) + CHR$(81) ' ↓(DownArr), PageDown
                GOSUB PageDown

            CASE CHR$(32), CHR$(0) + CHR$(59) '   Space Bar / F1 - data acquisition
                IF nsamp = ssper THEN '  has current acquisition already completed?
                    FOR mm = 0 TO 7
                        SCREEN , , mm, VS
                        LOCATE 3, 2: COLOR 14, 4
                        PRINT " Current data acquisition already completed, "
                        LOCATE 4, 2: PRINT " for another study use the F9-key "
                    NEXT
                    SCREEN , , VS, VS
```

```
                PLAY "L8 n37 n37 n37"
                GOTO DataAcq2 '       this study is done
            END IF
            IF PN >= 0 THEN '         check printer availability
                GOSUB DataAcqPrinterCheck
                IF ESC = 1 GOTO DataAcq2 '   ESC in DataAcqPrinterCheck
            END IF
            GOTO DataAcq3

        CASE CHR$(0) + CHR$(62) '     F4 - select/change Printer
            GOSUB DataAcqPrinterSelect
            IF PN >= 0 THEN GOSUB DataAcqPrinterCheck

        CASE CHR$(0) + CHR$(65) '     F7 - Save collected data
            GOSUB SaveData

        CASE CHR$(0) + CHR$(68) '     F10 - HELP Screen
            GOSUB DataAcqHelp

        CASE ELSE

    END SELECT

    GOTO DataAcq2


DataAcqFilenameError: '   incorrect data storage filename
    ESC = 8 '    set indicator (ESC=8) for DataAcqFilename subroutine
    RESUME DataAcq1

' ******************************************** DataAcq F-key ********************************************
DataAcqKey:
    LOCATE 25, 2
    COLOR 12, 0: PRINT "F1"; : COLOR 10, 3: PRINT "Start";
    COLOR 12, 0: PRINT "SpaceBar"; : COLOR 14, 3: PRINT "Continue";
    COLOR 12, 0: PRINT "F4"; : COLOR 11, 3: PRINT "Printer ";
    COLOR 12, 0: PRINT "PgUp "; CHR$(18); " PgDn"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "F7"; : COLOR 9, 3: PRINT "SaveData";
```

```
    COLOR 12, 0: PRINT "F9"; : COLOR 1, 3: PRINT "NewStudy";
    COLOR 12, 0: PRINT "F10"; : COLOR 15, 3: PRINT "Help";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "Exit";
RETURN


' @@@@@@@@@@@@  Data collection section starts here  @@@@@@@@@@@@@
DataAcq3:
    FOR mm = 0 TO 7
        SCREEN , , mm, VS
        ClearLine 1, 1, 1, 80, 7 '    erase Prompt & restore datafile$,
        ClearLine 2, 3, 2, 48, 7 '    in case SaveData was done
        ClearLine 3, 1, 4, 48, 7 '    after starting data acquisition
        LOCATE 1, 1: COLOR 14, 4: PRINT " Data file: "; datafile$; " "
    NEXT
    SCREEN , , VS, VS

    IF PN >= 0 THEN '          printer enabled?
        temp$ = MID$(STR$(PN + 1), 2)
        OPEN "LPT" + temp$ + ":" FOR OUTPUT AS #3 '              open printer
        IF nsamp = 0 THEN PRINT #3, DATE$; "  "; datafile$; "  "; ID$
    END IF '                      print date & ID, to start data acquisition

    OPEN datafile$ FOR APPEND AS #2 '   open data storage file

    GOSUB DataAcqCollect ' collect 1st round of data before timer trap starts

    FOR mm = 0 TO 7
        SCREEN , , mm, VS
        LOCATE 25, 2
        COLOR 12, 0: PRINT "^F10"; : COLOR 10, 3: PRINT "place Marker";
        COLOR 12, 0: PRINT "PgUp "; CHR$(24); " "; CHR$(25); " PgDn";
        COLOR 14, 3: PRINT "change display screen";
        COLOR 12, 0: PRINT "ESC or Enter";
        COLOR 0, 3: PRINT "stop acquisition ";
    NEXT
    SCREEN , , VS, VS
```

```
    ON TIMER(samp) GOSUB DataAcqCollect '    on timer(sampling interval) trap
    TIMER ON '               enable timer (trap to data collection subroutine)

DataAcq4:
    k$ = INKEY$
'                              ↑ or PgUp (Page Up) - previous visual screen
    IF k$ = CHR$(0) + CHR$(72) OR k$ = CHR$(0) + CHR$(73) THEN
        GOSUB PageUp
    END IF
'                                ↓ or PgDn (Page Down) - next visual screen
    IF k$ = CHR$(0) + CHR$(80) OR k$ = CHR$(0) + CHR$(81) THEN
        GOSUB PageDown
    END IF
'                                  ^F10 - insert an event Marker & time
    IF k$ = CHR$(0) + CHR$(103) THEN
        PLAY "L32 n37 n37 n37"
        PRINT #2, CHR$(13); "└═══"; CHR$(16); "    "; TIME$;
        IF PN >= 0 THEN
            CALL PrStat(PN, PS) '   get printer status byte
            IF PS = 0 THEN
                PRINT #3, CHR$(13); "└═══>"; "    "; TIME$; CHR$(13)
              ELSE BEEP
            END IF
        END IF
    END IF

DataAcq5: '  return point (from DataAcqCollect) when acquisition is completed
    IF k$ = CHR$(13) OR k$ = CHR$(27) THEN '  Enter or ESC - stop acquisition
        TIMER OFF '   stop sampling timer
        CLOSE '   close all opened files (#2-storage file, #3-printer output)
        IF nsamp < ssper THEN '    acquisition not completed (only suspended)
            FOR mm = 0 TO 7
                SCREEN , , mm, VS
                LOCATE 3, 2: COLOR 14, 4: PRINT " Data acquisition suspended "
            NEXT
            SCREEN , , VS, VS
        END IF
        GOTO DataAcq2
```

```
    END IF

    GOTO DataAcq4

' @@@@@@@@@@@@@  Data collection section stops here  @@@@@@@@@@@@@


' ***************************************************************************************************
' ***************************** Collect data using assembly subroutine ****************************
' ***************************************************************************************************
DataAcqCollect:
    nsamp = nsamp + 1
    FOR mm = 0 TO 7 '    show current sample number on screen
        SCREEN , , mm, VS
        LOCATE 4, 2: COLOR 1, 7: PRINT TIME$; "    "; nsamp
    NEXT

    PRINT #2, CHR$(13); TIME$; " "; '  make a new line & store time

    adn = 33 '          adn = no. of A/D readings
    FOR m = 0 TO chn '          **** NOTE: DataAcqCollect loop index is m ****
        IF flagS(m) > 0 THEN '        Is channel selected ?
            IF flagF(m) = 4 THEN '            Thermistors - special treatment
                GOSUB DataAcqThermistor '  is needed for EXP-RES board
              ELSE
                GOSUB DataAcqSample
                SELECT CASE flagF(m) '    m is DataAcqCollect loop index
                    CASE 8 '                                Thermocouples
                        GOSUB DataAcqThermocouple
                    CASE 2 '                                Linear
                        disp = poly(m, 1) * ad + poly(m, 0)
                    CASE 1 '                                Polynomial
                        disp = poly(m, 5) * ad ^ 5 + poly(m, 4) * ad ^ 4 + poly(m, 3) *
ad ^ 3 + poly(m, 2) * ad ^ 2 + poly(m, 1) * ad + poly(m, 0)
                    CASE ELSE
                        disp = ad
                END SELECT
            END IF
```

```
DataAcqCollect1: '          Which screen, row, column ?
            SELECT CASE m '    m is DataAcqCollect loop index
                CASE 0 TO 15
                    SCREEN , , 0, VS: rowx = 8: colx = col3
                CASE 16 TO 31
                    SCREEN , , 0, VS: rowx = -8: colx = col8
                CASE 32 TO 47
                    SCREEN , , 1, VS: rowx = -24: colx = col3
                CASE 48 TO 63
                    SCREEN , , 1, VS: rowx = -40: colx = col8
                CASE 64 TO 79
                    SCREEN , , 2, VS: rowx = -56: colx = col3
                CASE 80 TO 95
                    SCREEN , , 2, VS: rowx = -72: colx = col8
                CASE 96 TO 111
                    SCREEN , , 3, VS: rowx = -88: colx = col3
                CASE ELSE
                    SCREEN , , 3, VS: rowx = -104: colx = col8
            END SELECT
            LOCATE rowx + m, colx - 1: COLOR 10, 7 '              display data
            PRINT " "; : PRINT USING "######.##"; disp; : PRINT "  "
            PRINT #2, disp; '                                      store data

            printsum(m) = printsum(m) + disp ' sum for average/printer output

            IF nsamp = nsout THEN '                               display average
                vv = printsum(m) / ssout
                LOCATE rowx + m, colx + 11: COLOR 14
                PRINT USING "######.##"; vv; : PRINT " "
                IF flagS(m) > 1 THEN GOSUB PlotData '    plot data ?
                IF PN < 0 THEN printsum(m) = 0 '   if no printing
            END IF '                                  then reinitialize printsum
        END IF '      else reinitialize printsum below (after printer output)
    NEXT m
```

```
    IF nsamp = nsout THEN '       printer output
        nsout = nsout + ssout '    reset printer output count
        IF PN >= 0 THEN '      if printer no. < 0 then printer disabled
            GOSUB DataAcqPrinterCheck
            IF ESC = 1 THEN '         ESC pressed in DataAcqPrinterCheck
                FOR n = 0 TO chn: printsum(n) = 0: NEXT 'reinitialize printsum
                k$ = CHR$(27) '     force chr$(27) <Esc> as keyboard input
                RETURN DataAcq5 '   then go back to DataAcq menu (DataAcq5)
            END IF

            PRINT #3, TIME$ '         print time
            FOR n = 0 TO chn '        print average data
                IF flagS(n) > 0 THEN
                    PRINT #3, name$(n); "  ";
                    PRINT #3, USING "###"; n;
                    PRINT #3, "  ";
                    PRINT #3, USING "######.##"; printsum(n) / ssout;
                    PRINT #3, "  "; unit$(n)
                    printsum(n) = 0 '   reinitialize printer/average buffer
                END IF
            NEXT n
        END IF
    END IF


    IF nsamp = ssper THEN '    acquisition completed
        FOR mm = 0 TO 7
            SCREEN , , mm, VS
            LOCATE 3, 2: COLOR 14, 4: PRINT " Data acquisition completed "
        NEXT
        SCREEN , , VS, VS
        k$ = CHR$(27) '       force chr$(27) <Esc> as keyboard input
        PLAY "L8 n37 n37 n37"
        RETURN DataAcq5
    END IF

RETURN
```

```
' ***************************************************************************************************
' ************************************** Plot graphic data ******************************************
' ***************************************************************************************************
PlotData: '      NOTE: m is DataAcqCollect loop index
    nn = flagS(m) - 2
    SCREEN , , nn + 4, VS
    x = nsamp / ssper * 60 '          x-axis coordinate
    IF vv > plot(nn, 1) THEN '        y-axis coordinate
        y = 18
      ELSEIF vv < plot(nn, 0) THEN
        y = 0
      ELSE y = (vv - plot(nn, 0)) * (18 / (plot(nn, 1) - plot(nn, 0)))
    END IF
    LOCATE 23 - y, 18 + x: COLOR 15, 7: PRINT CHR$(233); '    plot point
    SCREEN , , VS, VS
RETURN

' ***************************************************************************************************
' ************************************** A/D sampling routine ***************************************
' ***************************************************************************************************
DataAcqSample: '   NOTE: m is DataAcqCollect loop index
    sum = 0
    FOR n = 1 TO adn '      adn = no. of A/D readings
        CALL Acq(gain(m), NOT coded(m), rad) '   assembly subroutine Acq
        sum = sum + rad
    NEXT n
    ad = sum / adn * res(m) '   average & A/D bit resolution
RETURN

' ***************************************************************************************************
' ********************************** Thermocouple handling routine **********************************
' ***************************************************************************************************
DataAcqThermocouple: '    NOTE: m is DataAcqCollect loop index
    ad = (ad / 206.2111) - 3.378 '    convert A/D data to voltage (mV)
    disp = poly(m, 5) * ad ^ 5 + poly(m, 4) * ad ^ 4 + poly(m, 3) * ad ^ 3 + poly(m, 2) *
ad ^ 2 + poly(m, 1) * ad + poly(m, 0)
RETURN DataAcqCollect1
```

```
' ************************************************************************************************
' ********************************** Thermistor handling routine ***********************************
' ************************************************************************************************
DataAcqThermistor: '   NOTE: m is DataAcqCollect loop index
    TMvolt = 4000 '          TMvolt = thermistor excitation voltage = 4000 mV

    adn = 3 '              adn = no. of A/D readings
    GOSUB DataAcqSample '   First, do a dummy A/D sample to charge up
    adn = 66 '             Thermistor (EXP-RES) board multiplexor
    GOSUB DataAcqSample '   Finally, ready to sample A/D data

    IF ad <= 0 THEN '    ad test : divide by 0 at TMvolt-(-sense)
        disp = 0 '                TM open may give -ad or ≈-100°C disp
        GOTO DataAcqThermistor1 '  TM short ^] ad≈3mV, ≈-60°C on EXP-RES board
    END IF
    mm = m - 96 '          here, ad becomes the voltage across +sense & -sense
    ad = .3 * ad + .5 * fil1(mm) + .2 * fil2(mm) '   filtering
    fil2(mm) = fil1(mm)
    fil1(mm) = ad
    minusSense = TMvolt * 23 / 24 - ad
    IF minusSense <= 0 THEN '        -sense test : TM open ^] LOG(0)=∞
        disp = 0: GOTO DataAcqThermistor1 '        TM short ^] LOG(-no.)=∞
    END IF
    ohm = minusSense / (TMvolt - minusSense) * 23000 '   Compute resistance
    ohm = LOG(ohm) '    resistance (in ohm) of thermistor

    disp = 1 / (poly(m, 3) + poly(m, 2) * ohm + poly(m, 1) * ohm ^ 3) + poly(m, 0) -
273.15

DataAcqThermistor1:
RETURN DataAcqCollect1


' ************************************************************************************************
' ************************************* Printer status check ***************************************
' ************************************************************************************************
DataAcqPrinterCheck:
    ESC = 0
```

```
    CALL PrStat(PN, PS) '       get printer status byte
    IF PS = 0 GOTO DataAcqPrinterCheck2 '     printer ready

    SCREEN , , VS, VS '   printer not ready
    COLOR 12, 7 '          open printer error message window (WindowOpen 5,25)
    GOSUB PrinterWindow
    LOCATE 7, 14:  PRINT "        DATA ACQUISITION PRINTER OUTPUT ERROR!
"
    LOCATE 13, 14: PRINT "   when printer is ready, press any key to continue    "
    LOCATE 15, 14: PRINT "         press Esc key to exit data acquisition          "
    BEEP
    COLOR 15, 4

    SELECT CASE PS
        CASE 32
            LOCATE 10, 17: PRINT " printer out-of-paper, or printer connector off "

        CASE 40
            LOCATE 10, 20: PRINT " printer out-of-paper and printer offline "

        CASE ELSE ' or PS = 8
            LOCATE 9, 25: PRINT " printer off or printer offline "
            COLOR 12, 7
            LOCATE 10, 26: PRINT " or general printer I/O error "
    END SELECT

DataAcqPrinterCheck1:
    k$ = INKEY$
        IF k$ = "" GOTO DataAcqPrinterCheck1
            WindowClose 5, 25 '     key pressed, restore screen first
        IF k$ = CHR$(27) THEN
            ESC = 1 '         ESC key pressed, return to data acquisition
            GOTO DataAcqPrinterCheck2
          ELSE GOTO DataAcqPrinterCheck '  assume printer ready now,
        END IF '                              try to print again

DataAcqPrinterCheck2: '   return to data acquisition
RETURN
```

```
' ************************************************************************************************
' ***************************** Data acquisition printer select routine ****************************
' ************************************************************************************************
DataAcqPrinterSelect: '              printer# 1, 2, 3
    number = PN + 1 ' but BIOS uses printer# 0, 1, 2 (PN)
    COLOR 15, 7
    GOSUB PrinterWindow '    open printer selection window (WindowOpen 5,25)
    COLOR 11, 7
    LOCATE 8, 16: PRINT "printer number 1, 2, or 3,  0 to disable printer"
DataAcqPrinterSelect1:
    LOCATE 10, 16: PRINT "printer number:   "
    GetInteger number, ESC, 1, 10, 32, 11, 7 '  count,row,col,fcolor,bcolor
    IF ESC = 1 GOTO DataAcqPrinterSelect2
    IF number > 3 THEN
        LOCATE 8, 16: COLOR 27: PRINT "printer number 1, 2, or 3,"
        BEEP
        GOTO DataAcqPrinterSelect1
    END IF
    PN = number - 1

DataAcqPrinterSelect2:
    WindowClose 5, 25

DataAcqPrinterSelect3:
    FOR mm = 0 TO 7
        SCREEN , , mm, VS
        LOCATE 3, 1: COLOR 11, 7
        IF PN >= 0 THEN
            PRINT "Using printer #"; PN + 1; ", ";
            COLOR 15: PRINT "F4";
            COLOR 11: PRINT " changes/disables printer"
          ELSE
            PRINT "printer disabled, use ";
            COLOR 15: PRINT "F4";
            COLOR 11: PRINT " to select a printer   "
        END IF
    NEXT
    SCREEN . , VS, VS
```

```
RETURN


' ************************************************************************************************************
' ******************************** Data acquisition storage filename ********************************
' ************************************************************************************************************
DataAcqFilename:
    WindowOpen 8, 24 '   open get filename window
    GOSUB FilenameWindow
    LOCATE 9, 32: COLOR 14, 3: PRINT " Data acquisition "

    IF ESC = 8 THEN '   there was a data acq. filename error:
        ESC = 0 '        could not open that file
        BEEP
        LOCATE 23, 13: COLOR 14, 4: PRINT "incorrect data storage filename"
        GOTO DataAcqFilename1 '     skip study ID remark entry
    END IF

    COLOR 1, 7
    LOCATE 11, 3: PRINT "For this study, you need to enter a "; : COLOR 14: PRINT
"required "; : COLOR 1: PRINT "data storage filename.  There"
    LOCATE 12, 3: PRINT "is also an optional study identification remark which may be
entered."
    LOCATE 14, 3: PRINT "The optional study ID remark are used to identify this current
study run,"
    LOCATE 15, 3: PRINT "e.g. 'J. Doe, run #10, immersion study in Arctic chamber at
0°C'."
    LOCATE 16, 3: PRINT "This ID remark may be 64 characters long.  A date need not
be entered."
    LOCATE 17, 3: PRINT "Today's date will be included automatically.  If no ID is
required simply"
    LOCATE 18, 3: PRINT "press the "; CHR$(17); "┘Enter or the ESC key."
    LOCATE 20, 3: PRINT "ID remark: ";
    GetLabel ID$, ESC, 64, 20, 14, 0, 7 '  count,row,col,fcolor,bcolor

    IF ESC = 1 THEN WindowClose 8, 24: GOTO DataAcqFilename2 '  ESC pressed
```

```
DataAcqFilename1:
    COLOR 1: ClearLine 11, 3, 22, 78, 7
    LOCATE 10, 3: PRINT "Enter a data storage filename for this study, The
recommended storage"
    LOCATE 11, 3: PRINT "filename format is "; : COLOR 14: PRINT "d:filename.ext"; :
COLOR 1: PRINT " (14 characters), 2 characters disk drive,"
    LOCATE 12, 3: PRINT "8 characters filename, 3 characters extension (excluding the
'.'). although"
    LOCATE 13, 3: PRINT "longer filename path can be used. Only the filename need to
be entered, the"
    LOCATE 14, 3: PRINT "disk drive and extension may be omitted. The disk drive must
be a ramdisk."
    LOCATE 15, 3: PRINT "Hard disk and floppy disk are too slow serve as a storage
device during"
    LOCATE 16, 3: PRINT "the data acquisition phase. The default ramdisk designation is
"; : COLOR 14: PRINT "d:"; : COLOR 1: PRINT ", other"
    LOCATE 17, 3: PRINT "designations may be entered. The standard data file extension
is "; : COLOR 15, 1: PRINT "D28"; : COLOR 1, 7: PRINT ","
    LOCATE 18, 3: PRINT "although any other may be used. However, it is recommended
that you"
    LOCATE 19, 3: PRINT "stay with the standard designation to avoid data file confusion
later on."
    LOCATE 20, 3: PRINT "If no extension is entered, then a "; : COLOR 15, 1: PRINT
"D28"; : COLOR 1, 7: PRINT " will be appended automatically."
    LOCATE 22, 3: PRINT "Filename: ";

    GetName datafile$, ESC, 64, 22, 13, 14, 7 '  count,row,col,fcolor,bcolor
    WindowClose 8, 24 '   close get filename window

    IF ESC = 1 GOTO DataAcqFilename2 '    ESC key pressed
```

```
    mm = 0: nn = 0 '    check if disk drive and file extension were entered
    FOR n = 1 TO LEN(datafile$)
        IF MID$(datafile$, n, 1) = ":" THEN mm = n '   disk: entered
        IF MID$(datafile$, n, 1) = "." THEN nn = n '   .ext entered
    NEXT
    IF mm = 0 THEN datafile$ = "d:" + datafile$ '  no disk drive, appended d:
    IF nn = 0 THEN datafile$ = datafile$ + ".D28" '   no .ext, append .D28
     FOR mm = 0 TO 7
        SCREEN , , mm, VS
        ClearLine 1, 1, 1, 80, 7
        LOCATE 1, 1: COLOR 14, 4: PRINT " Data file: "; datafile$; " "
    NEXT
    SCREEN , , VS, VS

DataAcqFilename2:
RETURN


' ***************************************************************************************************
' ************************************ DataAcq Help screen ************************************
' ***************************************************************************************************
DataAcqHelp:
    WindowOpen 8, 24 '    open HELP screen window

    LOCATE 8, 2: COLOR 12, 0: PRINT "PgUp "; CHR$(18); " PgDn"; : COLOR 4, 7:
PRINT " - change display screen          ";
    OR 15, 2: PRINT "Esc"; : COLOR 0, 7: PRINT " or "; : COLOR 15, 2: PRINT "F10"; :
COLOR 0, 7: PRINT " to exit HELP screen    "
    LOCATE 9, 2: PRINT "  "; : COLOR 12, 0: PRINT "F1"; : COLOR 10, 7: PRINT " -
Start the data acquisition routine (when channel setup is completed).  "
    LOCATE 10, 2: COLOR 12, 0: PRINT "Sp'B"; : COLOR 14, 7: PRINT " - Continue this
current study. The data acquisition procedure may be    "
    LOCATE 11, 2: PRINT "       temporarily suspended (by pressing ESC/F9). This key
restarts the    "
    LOCATE 12, 2: PRINT "       study from the exact point of suspension. All program
parameters are  "
    LOCATE 13, 2: PRINT "       restored. Data collected prior to the suspension are
preserved.      "
```

```
    LOCATE 14, 2: PRINT "  "; : COLOR 12, 0: PRINT "F4"; : COLOR 11, 7: PRINT " -
The data acquisition routine produces a hardcopy output on the printer."
    LOCATE 15, 2: PRINT "       Printer designation (number 1, 2 or 3) tells the program
where to send "
    LOCATE 16, 2: PRINT "       the printer data.  A warning message will be produced if
the designated"
    LOCATE 17, 2: PRINT "       printer is not available (this includes such conditions as
printer out-"
    LOCATE 18, 2: PRINT "       of-paper).  Printer may be disabled if a hardcopy output
is not needed."
    LOCATE 19, 2: PRINT "  "; : COLOR 12, 0: PRINT "F7"; : COLOR 9, 7: PRINT " -
Save the collected dataset on a disk.  If necessary, the acquisition    "
    LOCATE 20, 2: PRINT "       program may be suspended at any time to save the
(partial) dataset      "
    LOCATE 21, 2: PRINT "       already collected.  Use the Space Bar to continue the
program.          "
    LOCATE 22, 2: PRINT "  "; : COLOR 12, 0: PRINT "F9"; :COLOR 1, 7: PRINT " -
Prepare for a new study by restoring the setup screen.                  "
    LOCATE 23, 2: PRINT "  "; : COLOR 12, 0: PRINT "Esc"; : COLOR 0, 7: PRINT " -
Exit the current data acquisition run.                                  "

DataAcqHelp1:
    k$ = INKEY$
    IF k$ <> CHR$(27) AND k$ <> CHR$(0) + CHR$(68) GOTO DataAcqHelp1
    WindowClose 8, 24 '     close HELP screen window
RETURN
```

```
' ************************************************************************************************
' ************ Data Acquisition display screen: leave time, names & units only ************
' ************************************************************************************************
DataAcqScreen:
   FOR mm = 0 TO 3 '          Screens 0 - 3 (setup screens)
      SCREEN , , mm, VS
      LOCATE 6, col3
         COLOR 10, 7: PRINT "   Data     "; : COLOR 14: PRINT "Average"
      LOCATE 6, col8
         COLOR 10: PRINT "   Data     "; : COLOR 14: PRINT "Average"
      FOR n = 8 TO 23
         IF flagS(mm * 32 + n - 8) = 0 THEN '    left side 16 channels
            LOCATE n, col1: PRINT SPC(7); '   (23 - 8 = 15)
            LOCATE n, col3: PRINT SPC(27);
          ELSE '                                  if channel selected,
            LOCATE n, col3: PRINT SPC(19); '  retain Name & Unit
         END IF
         IF flagS(mm * 32 + n + 8) = 0 THEN '    right side 16 channels
            LOCATE n, col6: PRINT SPC(7); '   (23 + 8 = 31)
            LOCATE n, col8: PRINT SPC(27);
          ELSE '                                  if channel selected,
            LOCATE n, col8: PRINT SPC(19); '  retain Name & Unit
         END IF
      NEXT n
   NEXT mm
   flagVS = 3 '     if no data plotting, don't need to show plot screens
   FOR mm = 0 TO 3 '                    flagVS = last screen with information
      IF plot(mm, 2) > -1 THEN flagVS = flagVS + 1 '   need plotting?
   NEXT
   IF flagVS = 3 GOTO DataAcqScreen1 '   no data plotting
   FOR mm = 4 TO flagVS '          data plot screens
      SCREEN , , mm, VS
      GOSUB PlotScreen
      GOSUB TimePrompt
      LOCATE 1, 1: COLOR 14, 4: PRINT " Data file: "; datafile$; " "
   NEXT mm

DataAcqScreen1:
```

```
    SCREEN , , VS, VS
RETURN



' ***********************************************************************************************************
' ********************************** Data Acquisition exit routine **********************************
' ***********************************************************************************************************
DataAcqEnd: '  This is the only place to exit DataAcq and return to WhichFile
    FOR mm = 0 TO 3
        SCREEN , , mm, VS
        COLOR 1, 7
        LOCATE 6, col3: PRINT " Input    Factor   "
        LOCATE 6, col8: PRINT " Input    Factor   "
    NEXT

    SCREEN , , VS, VS
    COLOR 14, 7 ' open restoring setup screen message window (WindowOpen 1,7)
    GOSUB WaitWindow
    LOCATE 4, 29: PRINT "Restoring Setup Screens"
    ESC = 16 '          ESC=16 indicates LoadSetup after DataAcq, thus don't
    GOSUB LoadSetup2 ' erase plot screens. LoadSetup will close message window

DataAcqEnd1:
    DACQ = 0 '  reset data acquisition flag
    ESC = 0 '   reset ESC key pressed flag
    flagVS = 7 '  show all 8 screens again
RETURN
```

```
' ************************************************************************************************
' ******************** Up Arrow & Page Up (PgUp) keys handling routine ********************
' ************************************************************************************************
PageUp:
   SELECT CASE VS
      CASE 0
         VS = flagVS '   when already on screen 0, go to last screen
         ch = ch + 96 '  adjust channel no. to screen 3
      CASE 1, 2, 3
         VS = VS - 1 '   go to previous screen
         ch = ch - 32 '  adjust channel no.
      CASE 4, 5, 6, 7
         VS = VS - 1 '   go to previous screen
      CASE ELSE
   END SELECT
   SCREEN , , VS, VS
RETURN

' ************************************************************************************************
' ****************** Down Arrow & Page Down (PgDn) key handling routine ******************
' ************************************************************************************************
PageDown:
   SELECT CASE VS
      CASE flagVS, 7
         VS = 0 '        when already on last screen, go to screen 0
         ch = ch - 96 '  adjust screen no. to screen 0
      CASE 3, 4, 5, 6
         VS = VS + 1 '   advance to next screen
      CASE 0, 1, 2
         VS = VS + 1 '   advance to next screen
         ch = ch + 32 '  adjust channel no.
      CASE ELSE
   END SELECT
   SCREEN , , VS, VS
RETURN
```

```
' ##########################################################################
' ################  DISP128 - screen setup subroutines for AC128:  ###############
' ###########  DeviceName, ChannelSelect, InputGain, ConFactor, UnitLabel ###########
' ##########################################################################

' Biophysics Data Acquisition System ac128.20   900801   Stephen KW. Chang


DECLARE SUB WindowOpen (firstline AS INTEGER, lastline AS INTEGER) ' Subprogram
to open a message window
DECLARE SUB WindowClose (firstline AS INTEGER, lastline AS INTEGER) ' Subprogram
to close a message window
DECLARE SUB ClearLine (firstline AS INTEGER, firstcol AS INTEGER, lastline AS
INTEGER, lastcol AS INTEGER, bcolor AS INTEGER) ' Subprogram to clear a line or a
rectangle
DECLARE SUB GetInteger (number AS SINGLE, ESC AS INTEGER, count AS INTEGER,
row AS INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) '
Subprogram to input an integer number from keyboard
DECLARE SUB GetLabel (label$, ESC AS INTEGER, count AS INTEGER, row AS
INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) ' Subprogram to
input a label from keyboard
DECLARE SUB GetNumber (number AS SINGLE, ESC AS INTEGER, count AS
INTEGER, row AS INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER)
' Subprogram to input a number from keyboard

'          parameters passed to subprograms WindowOpen/WindowClose/ClearLine
' firstline    first line of screen to be saved/restored/cleared
' lastline     last line of screen to be saved/restored/cleared

'         parameters passed to subprogram ClearLine
' firstcol    first column to be cleared
' lastcol     last column to be cleared

'        parameters passed to subprograms GetLabel/GetNumber/GetInteger
' label$     label name
' temp$      temporary label name
' number     number entered in subprogram
' count      no. of characters in filename
```

```
' fcolor    display foreground color in subprogram
' bcolor    display background color in subprogram

REM $DYNAMIC

COMMON SHARED /A/ row AS INTEGER, rowt AS INTEGER, col AS INTEGER, colt AS
INTEGER, coltt AS INTEGER, ch AS INTEGER, cht AS INTEGER, chn AS INTEGER, che
AS INTEGER
COMMON SHARED /B/ col1 AS INTEGER, col2 AS INTEGER, col3 AS INTEGER, col4
AS INTEGER, col5 AS INTEGER, col6 AS INTEGER, col7 AS INTEGER, col8 AS
INTEGER, col9 AS INTEGER, col10 AS INTEGER
COMMON SHARED /C/ flagS() AS INTEGER, flagF() AS INTEGER
COMMON SHARED /D/ plot() AS SINGLE
COMMON SHARED /E/ NC AS INTEGER, ESC AS INTEGER, VS AS INTEGER
COMMON SHARED /F/ gain() AS INTEGER, res() AS SINGLE, poly() AS SINGLE
COMMON SHARED /G/ samp AS SINGLE, sout AS SINGLE, sper AS SINGLE
COMMON SHARED /H/ name$(), volt$(), unit$(), setupfile$
COMMON SHARED /I/ ScrStore() AS INTEGER

DIM flagS(127) AS INTEGER '   select flag
DIM flagF(127) AS INTEGER '   conversion factor flag
DIM gain(127) AS INTEGER '    gain code for subroutine Acq
DIM res(127) AS SINGLE '      data resolution
DIM poly(127, 5) AS SINGLE '  conversion factor array
DIM plot(3, 2) AS SINGLE '              data plot window max & min values
DIM name$(127), volt$(127), unit$(127) ' channel name, voltage level, unit name
DIM ScrStore(1999) AS INTEGER '         screen save/restore storage array
DIM number(5) AS SINGLE ' entered number from subprograms GetNumber & GetInteger
DIM number AS SINGLE '    number entered in subprogram

DIM m AS INTEGER, mm AS INTEGER '    dummy variables
DIM n AS INTEGER, nn AS INTEGER
DIM pp AS INTEGER, qq AS INTEGER
DIM vv AS SINGLE

REM $STATIC

SUB ChannelDisplaySetup128 STATIC
```

```
FOR mm = 0 TO 3 '   setup filename
    SCREEN , , mm, VS
    ClearLine 1, 1, 1, 80, 7
    ClearLine 2, 1, 4, 48, 7
    LOCATE 2, 1: COLOR 6: PRINT mm + 1 '       show page no.
    LOCATE 1, 2: COLOR 15, 4: PRINT " Setup file: "; setupfile$; " "
NEXT
IF VS > 3 THEN '                    if was viewing a plot screen
    IF plot(VS - 4, 2) >= 0 THEN '   and a non-empty plot screen,
        ch = plot(VS - 4, 2) '   then find the channel no. and
        S = FIX(ch / 32) '      go to that channel's setup
        nn = ch - 32 * VS
        IF nn < 16 THEN
            row = nn + 8: col = col2
          ELSE row = nn - 8: col = col7
        END IF
      ELSE '                            the viewed plot screen was empty
        VS = 0: ch = 0: row = 8: col = col2
    END IF
END IF
SCREEN , , VS, VS

NewColumn:
    SELECT CASE col
        CASE col1, col6
            GOSUB DeviceName '     device name
        CASE col2, col7
            GOSUB ChannelSelect '  A/D channel enable/disable
        CASE col3, col8
            GOSUB InputGain '      gain code/input voltage select
        CASE col4, col9
            GOSUB ConFactor '      TC/TM/Linear/Polynomial
        CASE ELSE ' (col5, col10)
            GOSUB UnitLabel '      unit label
    END SELECT

    GOTO NewColumn
```

```
' ************************************************************************************************
' ************************************ Device name selection ************************************
' ************************************************************************************************
DeviceName:
    SCREEN , , VS, VS
    COLOR 10, 0: LOCATE 4, 2: PRINT " Device Name Selection "
    LOCATE 6, col: PRINT " Name  "
    GOSUB NameUnitKey '                    Device Name F-key

DeviceName1:
    LOCATE row, col: COLOR 16, 3: PRINT name$(ch)
    rowt = row: colt = col: cht = ch
    NC = 0: ESC = 0

DeviceName2:
    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(27), CHR$(0) + CHR$(67) '   Esc & F9 - Screen Setup Completed
            ESC = 1 '           exit subprogram, return to WhichFile
            NC = 1

        CASE CHR$(0) + CHR$(73) '    PgUp (Page Up)
            GOSUB PageUp

        CASE CHR$(0) + CHR$(81) '    PgDn (Page Down)
            GOSUB PageDown

        CASE CHR$(0) + CHR$(72) '    ↑ (Up Arrow)
            GOSUB UpArrow

        CASE CHR$(0) + CHR$(80) '    ↓ (Down Arrow)
            GOSUB DownArrow
```

```
        CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
            NC = 1
            IF colt = col1 THEN
                col = col10: ch = ch + 16
              ELSE col = col5: ch = ch - 16
            END IF

        CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
            NC = 1
            IF colt = col1 THEN col = col2 ELSE col = col7


        CASE CHR$(0) + CHR$(59) '    F1 - Enter device name
            GOSUB SetupWindow1 '          (WindowOpen 7, 25)
            LOCATE mm + 2, 4: COLOR  , 3
            PRINT " Enter device name (7 characters max)          "
            temp$ = name$(ch)
            GetLabel temp$, ESC, 7, mm + 2, 43, 0, 3 '  count,row,col,fclr,bclr
            IF ESC = 1 THEN '       ESC pressed, no device name entered
              ELSEIF ESC = 4 THEN
                GOSUB SetupWindow2
                LOCATE mm + 5, 12
                PRINT "replace with device name from channel #      "
DeviceName3:
                GetInteger number, ESC, 3, mm + 5, 52, 0, 3
                IF ESC = 1 THEN '             ESC pressed, no replacement
                  ELSEIF number < 0 OR number > 127 THEN
                    LOCATE mm + 6, 16
                    COLOR 30, 4: PRINT "  channel no. out of range       "
                    BEEP
                    GOTO DeviceName3
                  ELSE name$(ch) = name$(number) ' must be valid replacement
                END IF
              ELSE name$(ch) = temp$
            END IF
            WindowClose 7, 25
```

```
        CASE CHR$(0) + CHR$(66) '    F8 - Remove Name
            name$(ch) = "none    "

        CASE CHR$(0) + CHR$(68) '    F10 - HELP Screen
            GOSUB DeviceNameHelp

        CASE ELSE
            GOTO DeviceName2

    END SELECT


    LOCATE rowt, colt
    IF name$(cht) = "none    " THEN
        COLOR , 7: PRINT SPC(7);
      ELSE COLOR 0: PRINT name$(cht)
    END IF

    IF NC = 0 THEN '          new column/new screen?
        GOTO DeviceName1 '          No
      ELSE ClearLine 4, 1, 4, 48, 7 '   Yes
        LOCATE 6, colt: COLOR 1: PRINT " Name  "
    END IF

    IF ESC = 1 THEN GOTO SetupDone ELSE RETURN
```

```
' ********** DeviceName/UnitLabel F-key **********
NameUnitKey:
    LOCATE 25, 2
    COLOR 12, 0: PRINT "F1"; : COLOR 14, 3: PRINT "Enter Name  ";
    COLOR 12, 0: PRINT "F8"; : COLOR 1, 3: PRINT "Remove Name  ";
    COLOR 12, 0: PRINT "PgUp"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT " "; CHR$(27); " "; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT " "; CHR$(24); " "; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT " "; CHR$(25); " "; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT " "; CHR$(26); " "; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "PgDn"; : COLOR 1, 3: PRINT "  ";
    COLOR 12, 0: PRINT "F9"; : COLOR 5, 3: PRINT "Done ";
    COLOR 12, 0: PRINT "F10"; : COLOR 15, 3: PRINT "HELP ";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "Exit ";
RETURN


' ********** DeviceName/UnitLabel Help **********
DeviceNameHelp:
    WindowOpen 8, 24 '    open HELP screen window
    LOCATE 8, 2: COLOR 0: PRINT " Device Name Selection "; : COLOR , 7: PRINT "
HELP screen            "; : COLOR 15, 2: PRINT "Esc"; : COLOR 0, 7: PRINT " or "; :
COLOR 15, 2: PRINT "F10"; : COLOR 0, 7: PRINT " to exit HELP screen "
    LOCATE 9, 2: PRINT "  "; : COLOR 12, 0: PRINT "F1"; : COLOR 14, 7: PRINT " - An
identification name can be entered for each channel.  This will      "
    LOCATE 10, 2: PRINT "       also be the name displayed on the data acquisition
display screen,     "
    LOCATE 11, 2: PRINT "       and printed on the printer output.  Maximum 7
characters are allowed.  "
    LOCATE 12, 2: PRINT "       All device names are filled with blank spaces to 7
characters long,    "
    LOCATE 13, 2: PRINT "       if less than 7 characters were entered.
        "


NameUnitHelp1: ' this part of message is shared by both DeviceName & UnitLabel

    LOCATE 14, 2: PRINT "  "; : COLOR 12, 0: PRINT "F8"; : COLOR 1, 7: PRINT " -
Remove a name/label.  To rename a channel, it is not necessary to      "
```

```
    LOCATE 15, 2: PRINT "       remove the old name/label first.  Simply use F1 to enter
a new name    "
    LOCATE 16, 2: PRINT "       (and overwrites the old channel name).
        "
    LOCATE 17, 2: PRINT "  "; : COLOR 12, 0: PRINT "F9"; : COLOR 5, 7: PRINT " -
When channel setup is completed, use F9 to exit to the main screen.    "
    LOCATE 18, 2: PRINT "       You will then have the opportunity to store this setup
screen, go to   "
    LOCATE 19, 2: PRINT "       another setup, start data acquisition, or save already
collected data. "
    LOCATE 20, 2: PRINT "  "; : COLOR 12, 0: PRINT "F10"; : COLOR 15, 7: PRINT " -
Whenever stuck, try F10 - HELP screen.                                    "
    LOCATE 21, 2: PRINT "  "; : COLOR 12, 0: PRINT "Esc"; : COLOR 0, 7: PRINT " -
Always exit to the previous screen.                                       "
    LOCATE 22, 2: PRINT "       Here, Esc is functionally equivalent to F9.
        "
    LOCATE 23, 2: PRINT "  "; : COLOR 12, 0: PRINT CHR$(27); CHR$(18); CHR$(26); :
COLOR 4, 7: PRINT " - arrows change row and column.    "; : COLOR 12, 0: PRINT
"PgUp"; : COLOR 4, 7: PRINT " "; : COLOR 12, 0: PRINT "PgDn"; : COLOR 4, 7: PRINT
" - change display screen.   "

NameUnitHelp2:
    k$ = INKEY$
    IF k$ <> CHR$(27) AND k$ <> CHR$(0) + CHR$(68) GOTO NameUnitHelp2
    WindowClose 8, 24 '   close HELP screen window
RETURN
```

```
' ************************************************************************************************
' ******************************** A/D channel parameter selection ********************************
' ************************************************************************************************
ChannelSelect:
    SCREEN , , VS, VS
    COLOR 10, 0: LOCATE 4, 2: PRINT " Channel Parameter Selection "
    LOCATE 6, col: PRINT " Ch"
    GOSUB ChannelSelectKey '              Channel Selection) F-key

ChannelSelect1:
    IF flagS(ch) > 1 THEN '       data plot channel
         COLOR 30, 0
       ELSEIF flagS(ch) = 1 THEN ' enabled channel
         COLOR 31, 0
      ELSE COLOR 23, 0 '          disabled channel
    END IF
    LOCATE row, col: PRINT USING "###"; ch
    rowt = row: colt = col: cht = ch
    NC = 0: ESC = 0

ChannelSelect2:
    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(27), CHR$(0) + CHR$(67) '   Esc & F9 - Screen Setup Completed
            ESC = 1 '           exit subprogram, return to WhichFile
            NC = 1

        CASE CHR$(0) + CHR$(73) '    PgUp (Page Up)
            GOSUB PageUp

        CASE CHR$(0) + CHR$(81) '    PgDn (Page Down)
            GOSUB PageDown

        CASE CHR$(0) + CHR$(72) '    ↑ (Up Arrow)
            GOSUB UpArrow
```

```
    CASE CHR$(0) + CHR$(80) '    ↓ (Down Arrow)
        GOSUB DownArrow

    CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
        NC = 1
        IF colt = col2 THEN col = col1 ELSE col = col6

    CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
        NC = 1
        IF colt = col2 THEN col = col3 ELSE col = col8

    CASE CHR$(0) + CHR$(59) '    F1 - acquisition Timing
        GOSUB TimeCycle

    CASE CHR$(0) + CHR$(61) '    F3 - Enable data plotting
        GOSUB PlotData

    CASE CHR$(0) + CHR$(62) '    F4 - Disable data plotting
        IF flagS(ch) > 1 THEN plot(flagS(ch) - 2, 2) = -1: flagS(ch) = 1

    CASE CHR$(0) + CHR$(63) '    F5 - Enable Channel
        IF flagS(ch) = 0 THEN flagS(ch) = 1: che = che + 1

    CASE CHR$(0) + CHR$(66) '    F8 - Disable Channel
        IF flagS(ch) > 1 THEN plot(flagS(ch) - 2, 2) = -1
        IF flagS(ch) > 0 THEN flagS(ch) = 0: che = che - 1

    CASE CHR$(0) + CHR$(68) '    F10 - HELP Screen
        GOSUB ChannelSelectHelp

    CASE ELSE
        GOTO ChannelSelect2

END SELECT
```

```
    IF flagS(cht) > 1 THEN
        COLOR 14, 7
      ELSEIF flagS(cht) = 1 THEN
        COLOR 15, 7
      ELSE COLOR 8, 7
    END IF
    LOCATE rowt, colt: PRINT USING "###"; cht

    IF NC = 0 THEN  '         new column/new screen?
        GOTO ChannelSelect1 '         No
      ELSE ClearLine 3, 1, 4, 48, 7 '  Yes (line 3 may have
        LOCATE 6, colt: COLOR 1: PRINT " Ch" '  no-printer-output warning)
    END IF

    IF ESC = 1 THEN GOTO SetupDone ELSE RETURN


' ************************************ Channel select  F-key ************************************
ChannelSelectKey:
    LOCATE 25, 2
    COLOR 12, 0: PRINT "F1"; : COLOR 11, 3: PRINT "Timing";
    COLOR 12, 0: PRINT "F3"; : COLOR 14, 3: PRINT "PlotData";
    COLOR 12, 0: PRINT "F4"; : COLOR 9, 3: PRINT "NoPlot";
    COLOR 12, 0: PRINT "F5"; : COLOR 10, 3: PRINT "Enable";
    COLOR 12, 0: PRINT "F8"; : COLOR 1, 3: PRINT "Disable ";
    COLOR 12, 0: PRINT "PgUp"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(27);
    COLOR 12, 0: PRINT CHR$(24);
    COLOR 12, 0: PRINT CHR$(25);
    COLOR 12, 0: PRINT CHR$(26); : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "PgDn"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "F9"; : COLOR 5, 3: PRINT "Done";
    COLOR 12, 0: PRINT "F10"; : COLOR 15, 3: PRINT "HELP";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "Exit";
RETURN
```

```
' @@@@@@  Select sampling interval, printer output cycle, & study period @@@@@@

TimeCycle:
    row = 16 '    force top display window, row will be restored later
    GOSUB SetupWindow1 '   (WindowOpen 7, 25)
    number(0) = samp
    number(1) = sout
    number(2) = sper
    ClearLine 3, 1, 3, 48, 7 '   line 3 may have "No printer output" warning


    COLOR 27: ClearLine 11, 16, 11, 60, 0 '    Sampling interval
TimeCycleSamp:
    LOCATE 2, 49: PRINT " Sampling interval ="; number(0); "sec "
'                          Thermistor sampling is the most time consuming
'                 requiring ≈ 50 msec.  There is a total of 32 TM channels.
'                 if all were enabled, they need 1.6 sec. of sampling time.
    IF che = 0 THEN m = chn + 1 ELSE m = che
    SELECT CASE m '   samp limits for new setup
        CASE IS < 21
            mm = 1 '    if < 21 enabled ch. then samp = 1
        CASE 21 TO 45
            mm = 2 '    if 21 to 45 enabled ch. then samp ≥ 2
        CASE 46 TO 86
            mm = 3 '    if 46 to 86 enabled ch. then samp ≥ 3
        CASE ELSE
            mm = 4 '    if 87 to 128 enabled ch. then samp ≥ 4
    END SELECT
    IF number(0) < mm THEN number(0) = mm
    COLOR 11: LOCATE 8, 16: PRINT " valid input range: "; mm; " - 86400  "
    LOCATE 10, 16, 1, 0, 8 '                        86400 sec = 24 hr
    PRINT " Enter sampling interval (in seconds)        "
    GetInteger number(0), ESC, 5, 10, 55, 11, 0 ' count,row,col,fcolor,bcolor
    IF ESC = 1 GOTO TimeCycle1
    ClearLine 2, 49, 2, 79, 0
    LOCATE 2, 49: PRINT " Sampling interval ="; number(0); "sec "
    IF number(0) < mm OR number(0) > 86400 THEN
        LOCATE 11, 16: COLOR 30, 4: PRINT " Sampling Interval out of range  "
```

```
        COLOR 28, 0: BEEP
        GOTO TimeCycleSamp
    END IF
    ClearLine 8, 16, 11, 60, 7


    COLOR 27: ClearLine 12, 16, 12, 56, 0 '    Average/Printer Output Cycle
TimeCycleSout:
    LOCATE 3, 49: PRINT " Printer cycle ="; number(1); "sec "
    IF number(0) < 10 THEN nn = 10 ELSE nn = number(0)
    IF sout = 0 THEN number(1) = nn '                default sout for new setup
    COLOR 11: LOCATE 9, 16: PRINT " valid input range:  "; nn; " - 86400  "
    LOCATE 11, 16, 1, 0, 8
    PRINT " Enter printer cycle (in seconds)          "
    GetInteger number(1), ESC, 5, 11, 51, 11, 0 ' count,row,col,fcolor,bcolor
    IF ESC = 1 GOTO TimeCycle1
    ClearLine 3, 49, 3, 79, 0
    LOCATE 3, 49: PRINT " Printer cycle ="; number(1); "sec "
    IF number(1) < nn OR number(1) > 86400 THEN
        LOCATE 12, 16: COLOR 30, 4: PRINT " Printer Cycle out of range       "
        COLOR 28, 0: BEEP
        GOTO TimeCycleSout
    END IF
    ClearLine 9, 16, 12, 56, 7


    COLOR 27: ClearLine 13, 16, 13, 55, 0 '    Study Period
TimeCycleSper:
    LOCATE 4, 49: PRINT " Study period ="; number(2); "min"
'                 Each enabled channel stores 10 bytes per sampling period
'                    (add another 10 bytes for TIME).  For example, with
'          80 enabled channels, 10*(80+1) bytes of data per sampling period.
    IF number(0) < 60 THEN mm = 1 ELSE mm = CINT(number(0) / 60 + 1)
    IF che = 0 THEN m = chn + 1 ELSE m = che
    nn = FIX(2048000 / (10 * (m + 1) * 60)) * number(0) '2048000=2MB on ram disk
    IF sper = 0 THEN number(2) = mm '                default sper for new setup
    COLOR 11: LOCATE 10, 16: PRINT " valid input range:  "; mm; " - "; nn; " "
    LOCATE 12, 16, 1, 0, 8
```

```
    PRINT " Enter study period (in minutes)          "
    GetInteger number(2), ESC, 5, 12, 50, 11, 0 ' count,row,col,fcolor,bcolor
    IF ESC = 1 GOTO TimeCycle1
    ClearLine 4, 49, 4, 79, 0
    LOCATE 4, 49: PRINT " Study period ="; number(2); "min "
    IF number(2) < mm OR number(2) > nn THEN
        LOCATE 13, 16: COLOR 30, 4: PRINT " Study Period out of range        "
        COLOR 28, 0: BEEP
        GOTO TimeCycleSper
    END IF

    samp = number(0) '   valid time/cycle parameters entered
    sout = number(1)
    sper = number(2)
    IF sout / 60 > sper THEN '   printer cycle > study period warning message
        LOCATE 3, 1: COLOR 12, 0
        PRINT " printer cycle > study period, NO printer output"
    END IF
TimeCycle1:
    row = rowt '   restore row
    ESC = 0 '      reset ESC, if set by GetInteger
    WindowClose 7, 25
    FOR mm = 0 TO 7 '   Update time/cycle information on all display screens
        SCREEN , , mm, VS
        COLOR 11: ClearLine 2, 49, 4, 79, 0
        LOCATE 2, 49: PRINT " Sampling interval ="; samp; "sec "
        LOCATE 3, 49: PRINT " Printer cycle ="; sout; "sec "
        LOCATE 4, 49: PRINT " Study period ="; sper; "min "
    NEXT
    SCREEN , , VS, VS '   return to working screen
RETURN
```

```
' @@@@@@@@  Enter plot data expected max & min values  @@@@@@@@

' for data plotting ch., flagS(ch) = 2 + plot#
'     e.g. 1st plot (plot#0) => flagS(ch) = 2
'          2nd plot (plot#1) => flagS(ch) = 3
'          3rd plot (plot#2) => flagS(ch) = 4
'          4th plot (plot#3) => flagS(ch) = 5
' for plot data parameters in plot():
'     plot(flagS(ch)-2,0) = min limit of plot window
'     plot(flagS(ch)-2,1) = max limit
'     plot(flagS(ch)-2,2) = ch.# (plot enabled) / -1 (plot disabled)

PlotData:
    GOSUB SetupWindow1 '  (WindowOpen 7, 25)
    pp = flagS(ch) '       store old flagS(ch), need to restore if ESC keyed
    IF flagS(ch) < 2 THEN
        qq = -1
        FOR n = 3 TO 0 STEP -1 '      find plot space, if found then
            IF plot(n, 2) = -1 THEN qq = n '   qq=next available plot space
        NEXT
        IF qq = -1 THEN '   all plot space taken
            COLOR 14, 4
            LOCATE mm + 2, 28: PRINT " No plot space available "
            LOCATE mm + 4, 28: PRINT " press any key to return "
            ESC = 1: BEEP
PlotData1:
            IF INKEY$ = "" GOTO PlotData1 ELSE GOTO PlotData6
        END IF
        flagS(ch) = 2 + qq '   qq = next plot space found
    END IF
    number(0) = plot(flagS(ch) - 2, 0) '   min
    number(1) = plot(flagS(ch) - 2, 1) '   max
    COLOR 14
    LOCATE mm + 1, 3: PRINT "Enter the maximum and minimum limits of the input data
"; : COLOR 15, 6: PRINT "(99999 < x < -99999)": COLOR 14, 7
    LOCATE mm + 2, 3: PRINT "to make optimun use of the plot window screen."
    LOCATE mm + 4, 3: PRINT "The window limits are usually the"
    LOCATE mm + 5, 3: PRINT "absolute maximum and minimum values"
```

```
    LOCATE mm + 6, 3: PRINT "that can be expected from the data"
    LOCATE mm + 7, 3: PRINT "collected on this channel."
    LOCATE mm + 3, 49: COLOR 14, 0: PRINT " Enter max and min values "
    LOCATE mm + 4, 54:          PRINT " max = "; : COLOR 7: PRINT number(1)
    LOCATE mm + 5, 54: COLOR 14: PRINT " min = "; : COLOR 7: PRINT number(0)
PlotData2:
    FOR n = 1 TO 0 STEP -1
        GetNumber number(n), ESC, 13, mm + 5 - n, 61, 14, 0
        IF ESC = 1 GOTO PlotData6 '   ESC=1 -- ESC - no input
        IF ESC = 2 GOTO PlotData4 '   ESC=2 -- keep remaining
        IF ESC = 4 THEN '             ESC=4 -- replace
            GOSUB SetupWindow2
            LOCATE mm + 5, 12
            PRINT "replace with device name from channel #      "
PlotData3:
            GetInteger number, ESC, 3, mm + 5, 52, 0, 3
            IF ESC = 1 THEN '    ESC pressed, no replacement
                GOTO PlotData6
              ELSEIF number < 0 OR number > 127 THEN
                LOCATE mm + 6, 16
                COLOR 30, 4: PRINT "  channel no. out of range       "
                BEEP
                GOTO PlotData3
              ELSEIF flagS(number) < 2 THEN
                LOCATE mm + 6, 16
                COLOR 30, 4: PRINT " ch."; number; "does not have data plot "
                BEEP
                GOTO PlotData3
              ELSE '                                  must be valid replacement
                plot(flagS(ch) - 2, 0) = plot(flagS(number) - 2, 0) '   min
                plot(flagS(ch) - 2, 1) = plot(flagS(number) - 2, 1) '   max
                GOTO PlotData5
            END IF
        END IF
```

```
        IF number(n) < -99999 OR number(n) > 99999 THEN
            LOCATE mm + 3, 3: COLOR 30, 4: PRINT " input number out-of-range
 "
            LOCATE mm + 1, 58: COLOR 31, 6: PRINT "(99999 < x < -99999)"
            BEEP: GOTO PlotData2
        END IF
    NEXT n

PlotData4:
    IF number(1) <= number(0) THEN
        LOCATE mm + 3, 3: COLOR 30, 4: PRINT " Incorrect max. and min. limits      "
        BEEP: GOTO PlotData2
    END IF
    plot(flagS(ch) - 2, 0) = number(0) '   min   valid min & max entered
    plot(flagS(ch) - 2, 1) = number(1) '   max

PlotData5:
    IF pp = 0 THEN che = che + 1 '   pp has the old flagS
    plot(flagS(ch) - 2, 2) = ch '    this plot enabled
    ESC = 0 '   indicate valid plot data or replacement

PlotData6:
    IF ESC = 1 THEN flagS(ch) = pp '   no data plot entry
    WindowClose 7, 25
RETURN

' @@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@
```

```
' ************************************** ChannelSelectHelp **************************************
ChannelSelectHelp:
    WindowOpen 8, 24 '     open HELP screen window
    GOTO ChannelSelectHelpPage1 '  display ChannelSelect HELP page 1

ChannelSelectHelp1:
    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(0) + CHR$(80), CHR$(0) + CHR$(81) '     ↑ or PgDn (Page Down)
            GOTO ChannelSelectHelpPage2

        CASE CHR$(0) + CHR$(72), CHR$(0) + CHR$(73) '     ↓ or PgUp (Page Up)
            GOTO ChannelSelectHelpPage1

        CASE CHR$(27), CHR$(0) + CHR$(68) '     Esc & F10 - leave HELP screen
            WindowClose 8, 24 '                 close HELP screen window
            RETURN

        CASE ELSE
            GOTO ChannelSelectHelp1

    END SELECT

ChannelSelectHelpPage1: ' ******************** ChannelSelect HELP page 1 ********************
    LOCATE 8, 2: COLOR 15, 0: PRINT " Channel Parameters "; : COLOR 0, 7: PRINT "
HELP screen                 "; : COLOR 15, 2: PRINT "Esc"; : COLOR 0, 7: PRINT " or "; :
COLOR 15, 2: PRINT "F10"; : COLOR 0, 7: PRINT " to exit HELP screen "
    LOCATE 9, 2: PRINT "  "; : COLOR 12, 0: PRINT "F1"; : COLOR 11, 7: PRINT " -
data acqusition Timing information.  Selection is provided for "; : COLOR 9: PRINT "<1>
": COLOR 11
    LOCATE 10, 2: PRINT "       data sampling interval, "; : COLOR 9: PRINT "<2>"; :
COLOR 11: PRINT " printer output cycle, and "; : COLOR 9: PRINT "<3>"; : COLOR 11:
PRINT " study period."
    LOCATE 11, 2: PRINT "       "; : COLOR , 0: PRINT "<1> Sampling interval - gives the
frequency (in seconds) of each round  "
    LOCATE 12, 2: COLOR , 7: PRINT "       "; : COLOR , 0: PRINT "       of data
```

```
collection.  Data (e.g. temperatures, voltages) from every"
    LOCATE 13, 2: COLOR , 7: PRINT "      "; : COLOR , 0: PRINT "      ENABLED
channels are collected once every sampling interval.  All "
    LOCATE 14, 2: COLOR , 7: PRINT "      "; : COLOR , 0: PRINT "      of the
collected data are stored in the computer, and the option  "
    LOCATE 15, 2: COLOR , 7: PRINT "      "; : COLOR , 0: PRINT "      for saving
collected data permanently onto a disk is provided.    "
    LOCATE 16, 2: COLOR , 7: PRINT "      <2> Printer cycle - indicates how often (in
seconds) the sampled data   "
    LOCATE 17, 2: PRINT "            are to be output to the printer, providing a hardcopy
record.  For"
    LOCATE 18, 2: PRINT "            each channel, the printer data represents an
average of all data  "
    LOCATE 19, 2: PRINT "            sampled since the last printout.  This average data
is also       "
    LOCATE 20, 2: PRINT "            displayed under the Data Acquisition screen column
'Average'.     "
    LOCATE 21, 2: PRINT "      "; : COLOR , 0: PRINT "<3> Study period - is the length
of the study session in minutes.       "
    LOCATE 22, 2: COLOR , 7: PRINT "  "; : COLOR 12, 0: PRINT CHR$(27); CHR$(18);
CHR$(26); : COLOR 4, 7: PRINT " - arrows change row and column.    "; : COLOR 12, 0:
PRINT "PgUp"; : COLOR 4, 7: PRINT " "; : COLOR 12, 0: PRINT "PgDn"; : COLOR 4, 7:
PRINT " - change display screen.   "
    LOCATE 23, 2: COLOR 0, 3: PRINT " ....MORE....   "; : COLOR 15: PRINT CHR$(25);
: COLOR 0: PRINT " or "; : COLOR 15: PRINT "PgDn"; : COLOR 0: PRINT " for next
screen    "; : COLOR 15: PRINT "Esc"; : COLOR 0: PRINT " or "; : COLOR 15: PRINT
"F10"; : COLOR 0: PRINT " to leave Help screen  "
    GOTO ChannelSelectHelp1

ChannelSelectHelpPage2: ' ***********  ChannelSelect HELP page 2  ***********
    LOCATE 8, 2: COLOR , 7: PRINT "  "; : COLOR 12, 0: PRINT "F3"; : COLOR 14, 7:
PRINT " - Plot data.  A time-course graphic plot of collected data which proceeds"
    LOCATE 9, 2: PRINT "       simultaneously with normal data display.  4 channels can
be assigned to"
    LOCATE 10, 2: PRINT "       be displayed on 4 display screens (one on each screen).
Data plot     "
    LOCATE 11, 2: PRINT "       selection, represented by "; : COLOR 14, 0: PRINT
"111"; : COLOR 14, 7: PRINT ", also automatically enables a channel.   "
```

```
    LOCATE 12, 2: COLOR , 7: PRINT "  "; : COLOR 12, 0: PRINT "F4"; : COLOR 9, 7:
PRINT " - Disable data plotting. This stops data plotting but leaves ch. enabled."
    LOCATE 13, 2: COLOR , 7: PRINT "  "; : COLOR 12, 0: PRINT "F5"; : COLOR 10, 7:
PRINT " - Enable a channel.  A brightened channel number "; : COLOR 15, 0: PRINT
"111"; : COLOR 10, 7: PRINT " indicates the        "
    LOCATE 14, 2: PRINT "        channel was enabled.  Only an enabled channel can
collect data.  Dis-  "
    LOCATE 15, 2: PRINT "        able all unused channels for a more efficient data
acquisition routine."
    LOCATE 16, 2: PRINT "  "; : COLOR 12, 0: PRINT "F8"; : COLOR 1, 7: PRINT " -
Disable a channel.  A darkened channel number "; : COLOR 7, 0: PRINT "111"; : COLOR
1, 7: PRINT " indicates a disabled "
    LOCATE 17, 2: PRINT "        channel.  This action disables both data collection and
data plotting. "
    LOCATE 18, 2: PRINT "  "; : COLOR 12, 0: PRINT "F9"; : COLOR 5, 7: PRINT " -
When channel setup is completed, use F9 to exit to the main screen.     "
    LOCATE 19, 2: PRINT "        You will then have the opportunity to store this setup
screen, go to    "
    LOCATE 20, 2: PRINT "        another setup, start data acquisition, or save already
collected data. "
    LOCATE 21, 2: PRINT "  "; : COLOR 12, 0: PRINT "F10"; : COLOR 15, 7: PRINT " -
Whenever stuck, try F10 -  HELP screen.                                    "
    LOCATE 22, 2: PRINT "  "; : COLOR 12, 0: PRINT "Esc"; : COLOR 0, 7: PRINT " -
Always exit to the previous screen.  Here, Esc is equivalent to F9.    ";
    LOCATE 23, 2: COLOR 0, 3: PRINT "                  "; : COLOR 15: PRINT CHR$(24); :
COLOR 0: PRINT " or "; : COLOR 15: PRINT "PgUp"; : COLOR 0: PRINT " for previous
screen     "; : COLOR 15: PRINT "Esc"; : COLOR 0: PRINT " or "; : COLOR 15: PRINT
"F10"; : COLOR 0: PRINT " to leave Help screen  "
    GOTO ChannelSelectHelp1
```

```
' ************************************************************************************************
' ***************************** Select gain code/input voltage range *****************************
' ************************************************************************************************
InputGain:
    SCREEN , , VS, VS
    COLOR 10, 0: LOCATE 4, 2: PRINT " Input Voltage (Gain Code) Selection "
    LOCATE 6, col: PRINT " Input"
    GOSUB InputGainKey '                    Input Voltage Range F-key

InputGain1:
    LOCATE row, col: COLOR 30, 4: PRINT volt$(ch)
    ClearLine 3, 1, 3, 48, 7 '    may need to erase message from InputGain3
    rowt = row: colt = col: cht = ch
    NC = 0: ESC = 0

InputGain2:
    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(27), CHR$(0) + CHR$(67) '   Esc & F9 - Screen Setup Completed
            ESC = 1 '           exit subprogram, return to WhichFile
            NC = 1

        CASE CHR$(0) + CHR$(73) '    PgUp (Page Up)
            GOSUB PageUp

        CASE CHR$(0) + CHR$(81) '    PgDn (Page Down)
            GOSUB PageDown

        CASE CHR$(0) + CHR$(72) '    ↑ (Up Arrow)
            GOSUB UpArrow

        CASE CHR$(0) + CHR$(80) '    ↓ (Down Arrow)
            GOSUB DownArrow
```

```
CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
    NC = 1
    IF colt = col3 THEN col = col2 ELSE col = col7

CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
    NC = 1
    IF colt = col3 THEN col = col4 ELSE col = col9


CASE CHR$(0) + CHR$(59) '           F1 = ± 5 V Input Range
    IF flagS(ch) = 0 THEN flagS(ch) = 1: che = che + 1
    gain(ch) = gain(ch) AND &H1F
    res(ch) = 2.44 '      gain() holds gain code in bits 5 & 6
    volt$(ch) = "± 5  V" '   and channel select in bits 0 to 4


CASE CHR$(0) + CHR$(60) '           F2 = ± 500 mV Input Range
    IF flagF(ch) >= 4 GOTO InputGain3
    IF flagS(ch) = 0 THEN flagS(ch) = 1: che = che + 1
    gain(ch) = &H20 OR (gain(ch) AND &H1F)
    res(ch) = .244
    volt$(ch) = "±500mV"


CASE CHR$(0) + CHR$(61) '           F3 = ± 50 mV Input Range
    IF flagF(ch) >= 4 GOTO InputGain3
    IF flagS(ch) = 0 THEN flagS(ch) = 1: che = che + 1
    gain(ch) = &H40 OR (gain(ch) AND &H1F)
    res(ch) = .0244
    volt$(ch) = "±50 mV"


CASE CHR$(0) + CHR$(62) '           F4 = ± 10 mV Input Range
    IF flagF(ch) >= 4 GOTO InputGain3
    IF flagS(ch) = 0 THEN flagS(ch) = 1: che = che + 1
    gain(ch) = &H60 OR (gain(ch) AND &H1F)
    res(ch) = .00488
    volt$(ch) = "±10 mV"
```

```
    CASE CHR$(0) + CHR$(66) '    F8 - Remove Input Voltage Range
        IF flagS(ch) > 0 THEN
            LOCATE 3, 1: COLOR 30, 4: PRINT "««";
            COLOR 14: PRINT "Channel enabled, cannot remove input voltage";
            COLOR 30: PRINT "»»"
            BEEP: GOTO InputGain2
          ELSE
            gain(ch) = gain(ch) AND &H1F '  remove only gain code,
            res(ch) = 0 '                   retain channel select
            volt$(ch) = "±      "
        END IF

    CASE CHR$(0) + CHR$(68) '    F10 - HELP Screen
        GOSUB InputGainHelp

    CASE ELSE
        GOTO InputGain2

END SELECT


SELECT CASE cht
    CASE 0 TO 15, 32 TO 47, 64 TO 79, 96 TO 111
        coltt = col2 '   use coltt, because we want to update
    CASE ELSE '          channel no. along with input gain select
        coltt = col7
END SELECT
```

```
    LOCATE rowt, coltt '     update channel selection
    IF flagS(cht) > 1 THEN
        COLOR 14, 7
      ELSEIF flagS(cht) = 1 THEN
        COLOR 15, 7
      ELSE COLOR 8, 7
    END IF
    PRINT USING "###"; cht; : PRINT " ";
    IF volt$(cht) <> "±      " THEN
        COLOR 14, 4: PRINT volt$(cht) '   update input gain select
      ELSE COLOR , 7: PRINT SPC(7);
    END IF

    IF NC = 0 THEN '          new column/new screen?
        GOTO InputGain1 '            No
      ELSE ClearLine 3, 1, 4, 48, 7 '   Yes (line 3 may have TM/TC ± 5V
        LOCATE 6, colt: COLOR 1: PRINT " Input" '     warning)
    END IF

    IF ESC = 1 THEN GOTO SetupDone ELSE RETURN


InputGain3: '             disallow TC/TM from other than ± 5V input selection
    LOCATE 3, 1: COLOR 30, 4: PRINT "««";
    COLOR 14: PRINT "Thermocouple/Thermistor must have ± 5V input";
    COLOR 30: PRINT "»»"
    BEEP
    GOTO InputGain2
```

```
' ***************************** Gain Code/Input Voltage Range  F-key ****************************
InputGainKey:
    LOCATE 25, 2
    COLOR 12, 0: PRINT "F1"; : COLOR 14, 3: PRINT "±5 V";
    COLOR 12, 0: PRINT "F2"; : COLOR 14, 3: PRINT "±500 mV";
    COLOR 12, 0: PRINT "F3"; : COLOR 14, 3: PRINT "±50 mV";
    COLOR 12, 0: PRINT "F4"; : COLOR 14, 3: PRINT "±10 mV";
    COLOR 12, 0: PRINT "F8"; : COLOR 1, 3: PRINT "Remove ";
    COLOR 12, 0: PRINT "PgUp"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(27); " "; CHR$(24); " "; CHR$(25); " "; CHR$(26); :
COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "PgDn"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "F9"; : COLOR 5, 3: PRINT "Done";
    COLOR 12, 0: PRINT "F10"; : COLOR 15, 3: PRINT "HELP";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "Exit ";
RETURN


' ***************************************** InputGainHelp ****************************************
InputGainHelp:
    WindowOpen 8, 24 '   open HELP screen window

    LOCATE 8, 2: COLOR 14, 4: PRINT " Input Voltage (Gain Code) "; : COLOR 0, 7:
PRINT " HELP screen          "; : COLOR 15, 2: PRINT "Esc"; : COLOR 0, 7: PRINT " or ";
: COLOR 15, 2: PRINT "F10"; : COLOR 0, 7: PRINT " to exit HELP screen "
    LOCATE 9, 2: PRINT "  "; : COLOR 12, 0: PRINT "F1"; : COLOR 14, 7: PRINT " -
Input voltage range ± 5 volts(V).  Resolution = 2.44 millivolts (mV).  "
    LOCATE 10, 2: PRINT "  "; : COLOR 12, 0: PRINT "F2"; : COLOR 14, 7: PRINT " -
Input voltage range ± 500 mV.  Resolution = 0.244 mV.                  "
    LOCATE 11, 2: PRINT "  "; : COLOR 12, 0: PRINT "F3"; : COLOR 14, 7: PRINT " -
Input voltage range ± 50 mV.  Resolution = 24.4 microvolts (µV).       "
    LOCATE 12, 2: PRINT "  "; : COLOR 12, 0: PRINT "F4"; : COLOR 14, 7: PRINT " -
Input voltage range ± 10 mV.  Resolution = 4.88 µV.                    "
    LOCATE 13, 2: PRINT "          "; : COLOR 15, 6: PRINT "NOTE: for this data
acquisition system, almost all input should be a"; : COLOR 6, 7: PRINT "    "
    LOCATE 14, 2: PRINT "          "; : COLOR 15, 6: PRINT "± 5 V range selection."; :
COLOR 6, 7: PRINT "  When "; : COLOR 12, 0: PRINT "F1"; : COLOR 6, 7: PRINT " - ";
: COLOR 12, 0: PRINT "F4"; : COLOR 6, 7: PRINT _
```

```
" are selected, the corresponding    "
    LOCATE 15, 2: PRINT "        channel is also automatically enabled, which may be
manually disabled  "
    LOCATE 16, 2: PRINT "        later, if necessary.
      "
    LOCATE 17, 2: PRINT "  "; : COLOR 12, 0: PRINT "F8"; : COLOR 1, 7: PRINT " -
Remove the selected input voltage range.  "; : COLOR 15, 1: PRINT "Note:"; : COLOR 1,
7: PRINT " the input voltage range"
    LOCATE 18, 2: PRINT "        of an enabled and/or plotting channel can not be
removed.                "
    LOCATE 19, 2: PRINT "  "; : COLOR 12, 0: PRINT "F9"; : COLOR 5, 7: PRINT " -
When channel setup is completed, use F9 to exit to the main screen.     "
    LOCATE 20, 2: PRINT "        You may then store this setup screen, or start data
acquisition.        "
    LOCATE 21, 2: PRINT "  "; : COLOR 12, 0: PRINT "F10"; : COLOR 15, 7: PRINT " -
Whenever stuck, try F10 - HELP screen.                                    "
    LOCATE 22, 2: PRINT "  "; : COLOR 12, 0: PRINT "Esc"; : COLOR 0, 7: PRINT " -
Always exit to the previous screen.  Here, Esc is equivalent to F9.    "
    LOCATE 23, 2: PRINT "  "; : COLOR 12, 0: PRINT CHR$(27); CHR$(18); CHR$(26); :
COLOR 4, 7: PRINT " - arrows change row and column.     "; : COLOR 12, 0: PRINT
"PgUp"; : COLOR 4, 7: PRINT " "; : COLOR 12, 0: PRINT "PgDn"; : COLOR 4, 7: PRINT
" - change display screen.    "

InputGainHelp1:
    k$ = INKEY$
    IF k$ <> CHR$(27) AND k$ <> CHR$(0) + CHR$(68) GOTO InputGainHelp1
    WindowClose 8, 24 '    close HELP screen window
    RETURN
```

```
' ************************************************************************************************
' ******************************  A/D channel conversion factor  *********************************
' ************************************************************************************************
ConFactor:
    SCREEN , , VS, VS
    COLOR 10, 0: LOCATE 4, 2: PRINT " Conversion Factor Selection "
    LOCATE 6, col: PRINT "   Factor   "
    GOSUB ConFactorKey '              Conversion Factor F-key

ConFactor1:
    ClearLine 1, 1, 1, 80, 7
    ClearLine 2, 1, 2, 48, 7
    LOCATE row, col: COLOR 31, 1
    SELECT CASE flagF(ch)
        CASE 8
            PRINT "thermocouple"
            LOCATE 1, 1
            COLOR 15, 3: PRINT poly(ch, 5); : COLOR 8: PRINT "x^5 +";
            COLOR 15: PRINT poly(ch, 4); : COLOR 8: PRINT "x^4 +";
            COLOR 15: PRINT poly(ch, 3); : COLOR 8: PRINT "x^3 +";
            COLOR 15: PRINT poly(ch, 2); : COLOR 8: PRINT "x^2 +";
            COLOR 15: PRINT poly(ch, 1); : COLOR 8: PRINT "x +";
            COLOR 15: PRINT poly(ch, 0)
        CASE 4
            PRINT " thermistor "
            LOCATE 1, 1
            COLOR 7, 1: PRINT " T = 1 / ["; : COLOR 15: PRINT poly(ch, 3);
            COLOR 7: PRINT " + "; : COLOR 15: PRINT poly(ch, 2);
            COLOR 7: PRINT " * (lnR) + "; : COLOR 15: PRINT poly(ch, 1);
            COLOR 7: PRINT " * (lnR)^3] + "; : COLOR 15: PRINT poly(ch, 0)
        CASE 2
            PRINT "   linear   "
            LOCATE 1, 1
            COLOR 15, 6: PRINT poly(ch, 1); : COLOR 7: PRINT " x + ";
            COLOR 15: PRINT poly(ch, 0)
```

```
        CASE 1
            PRINT " polynomial "
            LOCATE 1, 1
            COLOR 15, 2: PRINT poly(ch, 5); : COLOR 8: PRINT "x^5 +";
            COLOR 15: PRINT poly(ch, 4); : COLOR 8: PRINT "x^4 +";
            COLOR 15: PRINT poly(ch, 3); : COLOR 8: PRINT "x^3 +";
            COLOR 15: PRINT poly(ch, 2); : COLOR 8: PRINT "x^2 +";
            COLOR 15: PRINT poly(ch, 1); : COLOR 8: PRINT "x +";
            COLOR 15: PRINT poly(ch, 0)
        CASE ELSE
            PRINT "     1       "
            LOCATE 1, 2: COLOR 15, 4: PRINT " Setup file: "; setupfile$; " "
            LOCATE 2, 1: COLOR 6, 7: PRINT VS + 1 '    show page no.
    END SELECT
    rowt = row: colt = col: cht = ch
    NC = 0: ESC = 0

ConFactor2:
    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(27), CHR$(0) + CHR$(67) '   Esc & F9 - Screen Setup Completed
            ESC = 1 '           exit subprogram, return to WhichFile
            NC = 1

        CASE CHR$(0) + CHR$(73) '    PgUp (Page Up)
            GOSUB PageUp

        CASE CHR$(0) + CHR$(81) '    PgDn (Page Down)
            GOSUB PageDown

        CASE CHR$(0) + CHR$(72) '    ↑ (Up Arrow)
            GOSUB UpArrow

        CASE CHR$(0) + CHR$(80) '    ↓ (Down Arrow)
            GOSUB DownArrow
```

```
CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
    NC = 1
    IF colt = col4 THEN col = col3 ELSE col = col8

CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
    NC = 1
    IF colt = col4 THEN col = col5 ELSE col = col10


CASE CHR$(0) + CHR$(59) '    F1 - Thermocouples
    IF ch > 95 THEN
        LOCATE 3, 1: COLOR 14, 4 '                check for TM channels
        PRINT "channel "; : COLOR 30: PRINT "96 - 127 ";
        COLOR 14: PRINT "are usually thermistor channels";
    END IF
    GOSUB SetupWindow1 '    (WindowOpen 7, 25)
    GOSUB ConFactorWindowMessage
    LOCATE mm + 1, 3: COLOR 8, 3: PRINT " THERMOCOUPLE equation:    ";
        COLOR 15: PRINT "A"; : COLOR 8: PRINT " x^5 + ";
        COLOR 15: PRINT "B"; : COLOR 8: PRINT " x^4 + ";
        COLOR 15: PRINT "C"; : COLOR 8: PRINT " x^3 + ";
        COLOR 15: PRINT "D"; : COLOR 8: PRINT " x^2 + ";
        COLOR 15: PRINT "E"; : COLOR 8: PRINT " x + ";
        COLOR 15: PRINT "F "
    LOCATE mm + 2, 3: COLOR 8: PRINT " Enter coefficients: ";
        COLOR 15: PRINT "A, B, C, D, E, ";
        COLOR 8: PRINT "and "; : COLOR 15: PRINT "F "
    IF flagF(ch) = 8 THEN
        FOR n = 0 TO 5: number(n) = poly(ch, n): NEXT
      ELSE
        number(5) = 0 '              set thermocouple coefficients
        number(4) = -.000355009# '   using NBS (monograph 125)
        number(3) = .022181644# '    type T, copper-constantan,
        number(2) = -.61954869# '    thermocouple equation
        number(1) = 25.661297# '     (in mV <not μV> input form)
        number(0) = 0
    END IF
 LOCATE mm + 2, 50:              PRINT " A = "; : COLOR 8: PRINT number(5)
```

```
        LOCATE mm + 3, 50: COLOR 15: PRINT " B = "; : COLOR 8: PRINT number(4)
        LOCATE mm + 4, 50: COLOR 15: PRINT " C = "; : COLOR 8: PRINT number(3)
        LOCATE mm + 5, 50: COLOR 15: PRINT " D = "; : COLOR 8: PRINT number(2)
        LOCATE mm + 6, 50: COLOR 15: PRINT " E = "; : COLOR 8: PRINT number(1)
        LOCATE mm + 7, 50: COLOR 15: PRINT " F = "; : COLOR 8: PRINT number(0)
            FOR n = 5 TO 0 STEP -1
                GetNumber number(n), ESC, 13, mm + 7 - n, 55, 8, 3
                IF ESC = 1 GOTO ConFactor5TC '  ESC=1 -- ESC - no input
                IF ESC = 2 GOTO ConFactor3TC '  ESC=2 -- keep remaining
                IF ESC = 4 THEN '               ESC=4 -- replace
                    GOSUB ConFactorReplace
                    IF ESC = 1 GOTO ConFactor5TC '  ESC -- no replacement
                    GOTO ConFactor4TC '     (ESC=4) replaced
                END IF
            NEXT n
ConFactor3TC:
            FOR n = 0 TO 5: poly(ch, n) = number(n): NEXT
ConFactor4TC:
            flagF(ch) = 8
            gain(ch) = gain(ch) AND &H1F '   also automatically
            res(ch) = 2.44 '    selects ± 5V input voltage range
            volt$(ch) = "± 5  V"
ConFactor5TC:
            WindowClose 7, 25


        CASE CHR$(0) + CHR$(60) '    F2 - Thermistors
            IF ch < 96 THEN
                LOCATE 3, 1: COLOR 14, 4 '        check for TM channel location
                PRINT "Thermistors must use channels ";
                COLOR 30: PRINT "96 - 127"
            END IF
            GOSUB SetupWindow1 '    (WindowOpen 7, 25)
            GOSUB ConFactorWindowMessage
            LOCATE mm + 1, 3: COLOR 7, 1: PRINT " THERMISTOR {Steinhart-Hart} ";
                PRINT "equation: T = 1 / [";
                COLOR 15: PRINT "a"; : COLOR 7: PRINT " + ";
                COLOR 15: PRINT "b"; : COLOR 7: PRINT "(lnR) + ";
```

```
            COLOR 15: PRINT "c"; : COLOR 7: PRINT "(lnR)^3] + ";
            COLOR 15: PRINT "d  ": COLOR 7
        LOCATE mm + 2, 3: PRINT " lnR = natural log of R (Ω ohms), ";
            PRINT "T = temperature in °C (converted from °K) "
        LOCATE mm + 3, 40: COLOR 7: PRINT " Enter coefficients: ";
            COLOR 15: PRINT "a, b, c, d "
        number(5) = 0: number(4) = 0
        IF flagF(ch) = 4 THEN
            FOR n = 0 TO 3: number(n) = poly(ch, n): NEXT
          ELSE
            number(3) = 1.45937E-03 '   a   set Steinhart-Hart
            number(2) = 2.39735E-04 '   b   thermistor equation
            number(1) = 9.47611E-08 '   c   coefficients
            number(0) = 0 '             d   (d is an offset)
        END IF
    LOCATE mm + 4, 54:              PRINT " a = "; : COLOR 7: PRINT number(3)
    LOCATE mm + 5, 54: COLOR 15: PRINT " b = "; : COLOR 7: PRINT number(2)
    LOCATE mm + 6, 54: COLOR 15: PRINT " c = "; : COLOR 7: PRINT number(1)
    LOCATE mm + 7, 54: COLOR 15: PRINT " d = "; : COLOR 7: PRINT number(0)
        FOR n = 3 TO 0 STEP -1
            GetNumber number(n), ESC, 13, mm + 7 - n, 59, 7, 1
            IF ESC = 1 GOTO ConFactor5TM '  ESC=1 -- ESC - no input
            IF ESC = 2 GOTO ConFactor3TM '  ESC=2 -- keep remaining
            IF ESC = 4 THEN '               ESC=4 -- replace
                GOSUB ConFactorReplace
                IF ESC = 1 GOTO ConFactor5TM '  ESC -- no replacement
                GOTO ConFactor4TM '         (ESC=4) replaced
            END IF
        NEXT n
ConFactor3TM:
        FOR n = 0 TO 5: poly(ch, n) = number(n): NEXT
ConFactor4TM:
        flagF(ch) = 4
        gain(ch) = gain(ch) AND &H1F '    also automatically
        res(ch) = 2.44 '    selects ± 5V input voltage range
        volt$(ch) = "± 5  V"
ConFactor5TM:
        WindowClose 7, 25
```

```
        CASE CHR$(0) + CHR$(61) '    F3 - Linear Conversion Factors
            IF ch > 95 THEN
                LOCATE 3, 1: COLOR 14, 4 '                  check for TM channels
                PRINT "channel "; : COLOR 30: PRINT "96 - 127 ";
                COLOR 14: PRINT "are usually thermistor channels";
            END IF
            GOSUB SetupWindow1 '   (WindowOpen 7, 25)
            GOSUB ConFactorWindowMessage
            LOCATE mm + 1, 3: COLOR 7, 0: PRINT " LINEAR CONVERSION:  ";
                COLOR 15: PRINT "a"; : COLOR 7: PRINT " x + ";
                COLOR 15: PRINT "b      ";
                COLOR 7: PRINT "Enter linear factors: ";
                COLOR 15: PRINT "a, b "
            number(5) = 0: number(4) = 0: number(3) = 0: number(2) = 0
            IF flagF(ch) = 2 THEN
                number(1) = poly(ch, 1): number(0) = poly(ch, 0)
              ELSE
                number(1) = 1: number(0) = 0 '  identity linear coefficients
            END IF
         LOCATE mm + 3, 45: PRINT " a = "; : COLOR 7: PRINT number(1)
         LOCATE mm + 4, 45: COLOR 15: PRINT " b = "; : COLOR 7: PRINT number(0)
            FOR n = 1 TO 0 STEP -1
                GetNumber number(n), ESC, 13, mm + 4 - n, 50, 7, 0
                IF ESC = 1 GOTO ConFactor5LN '  ESC=1 -- ESC - no input
                IF ESC = 2 GOTO ConFactor3LN '  ESC=2 -- keep remaining
                IF ESC = 4 THEN '                ESC=4 -- replace
                    GOSUB ConFactorReplace
                    IF ESC = 1 GOTO ConFactor5LN '  ESC -- no replacement
                    GOTO ConFactor4LN '      (ESC=4) replaced
                END IF
            NEXT n
ConFactor3LN:
            FOR n = 0 TO 5: poly(ch, n) = number(n): NEXT
ConFactor4LN:
            flagF(ch) = 2
ConFactor5LN:
            WindowClose 7, 25
```

```
        CASE CHR$(0) + CHR$(62) '    F4 - Polynomial Conversion Factors
            IF ch > 95 THEN
                LOCATE 3, 1: COLOR 14, 4 '                  check for TM channels
                PRINT "channel "; : COLOR 30: PRINT "96 - 127 ";
                COLOR 14: PRINT "are usually thermistor channels";
            END IF
            GOSUB SetupWindow1 '   (WindowOpen 7, 25)
            GOSUB ConFactorWindowMessage
            LOCATE mm + 1, 3: COLOR 8, 2: PRINT " POLYNOMIAL conversion
equation: ";
                COLOR 15: PRINT "A"; : COLOR 8: PRINT "x^5 + ";
                COLOR 15: PRINT "B"; : COLOR 8: PRINT "x^4 + ";
                COLOR 15: PRINT "C"; : COLOR 8: PRINT "x^3 + ";
                COLOR 15: PRINT "D"; : COLOR 8: PRINT "x^2 + ";
                COLOR 15: PRINT "E"; : COLOR 8: PRINT "x + ";
                COLOR 15: PRINT "F "
            LOCATE mm + 2, 3: COLOR 8: PRINT " Enter coefficients: ";
                COLOR 15: PRINT "A, B, C, D, E, ";
                COLOR 8: PRINT "and "; : COLOR 15: PRINT "F "
            IF flagF(ch) = 1 THEN
                FOR n = 0 TO 5: number(n) = poly(ch, n): NEXT
              ELSE
                FOR n = 0 TO 5: number(n) = 0: NEXT 'zero out all coefficients
            END IF
          LOCATE mm + 2, 52:                PRINT " A = "; : COLOR 8: PRINT numbe( 5)
          LOCATE mm + 3, 52: COLOR 15: PRINT " B = "; : COLOR 8: PRINT number( 4)
          LOCATE mm + 4, 52: COLOR 15: PRINT " C = "; : COLOR 8: PRINT number(3)
          LOCATE mm + 5, 52: COLOR 15: PRINT " D = "; : COLOR 8: PRINT number(2)
          LOCATE mm + 6, 52: COLOR 15: PRINT " E = "; : COLOR 8: PRINT number(1)
          LOCATE mm + 7, 52: COLOR 15: PRINT " F = "; : COLOR 8: PRINT number(0)
            FOR n = 5 TO 0 STEP -1
                GetNumber number(n), ESC, 13, mm + 7 - n, 57, 8, 2
                IF ESC = 1 GOTO ConFactor5PY '  ESC=1 -- ESC - no input
                IF ESC = 2 GOTO ConFactor3PY '  ESC=2 -- keep remaining
                IF ESC = 4 THEN '                ESC=4 -- replace
                    GOSUB ConFactorReplace
                    IF ESC = 1 GOTO ConFactor5PY '  ESC -- no replacement
                    GOTO ConFactor4PY '       (ESC=4) replaced
```

```
                END IF
            NEXT n
ConFactor3PY:
            FOR n = 0 TO 5: poly(ch, n) = number(n): NEXT
ConFactor4PY:
            flagF(ch) = 1
ConFactor5PY:
            WindowClose 7, 25


        CASE CHR$(0) + CHR$(66) '    F8 - Remove all conversion factors
            flagF(ch) = 0
            FOR n = 0 TO 5: poly(ch, n) = 0: NEXT

        CASE CHR$(0) + CHR$(68) '    F10 - HELP Screen
            GOSUB ConFactorHelp

        CASE ELSE
            GOTO ConFactor2

    END SELECT


    IF flagF(cht) = 8 OR flagF(cht) = 4 THEN
        SELECT CASE cht
            CASE 0 TO 15, 32 TO 47, 64 TO 79, 96 TO 111
                coltt = col3 ' use coltt to update Input Voltage
            CASE ELSE '        TC/TM selection will automatically select ± 5V
                coltt = col8
        END SELECT
        LOCATE rowt, coltt: COLOR 14, 4: PRINT volt$(cht)
    END IF
```

```
    LOCATE rowt, colt: COLOR 15, 1 '    now update
    SELECT CASE flagF(cht)
        CASE 8
            PRINT "thermocouple"
        CASE 4
            PRINT " thermistor "
        CASE 2
            PRINT "   linear   "
        CASE 1
            PRINT " polynomial "
        CASE ELSE
            COLOR , 7: PRINT SPC(12);
    END SELECT

    IF NC = 0 THEN '          new column/new screen?
        GOTO ConFactor1 '           No
      ELSE ClearLine 1, 1, 1, 80, 7 '   Yes
        ClearLine 2, 1, 4, 48, 7
        LOCATE 6, colt: COLOR 1: PRINT "   Factor   "
        LOCATE 1, 2: COLOR 15, 4: PRINT " Setup file: "; setupfile$; " "
        LOCATE 2, 1: COLOR 6, 7: PRINT VS + 1 '    show page no.
    END IF

    IF ESC = 1 THEN GOTO SetupDone ELSE RETURN

' ******************************** ConFactor input window message ********************************
ConFactorWindowMessage:
    COLOR 15, 4
    LOCATE mm + 4, 3: PRINT " limit each input to 13 digits "
    LOCATE mm + 5, 3: PRINT " maximum, e.g. -2.456789E+23   "
    LOCATE mm + 6, 3: PRINT " range: -3.37E+38 to 3.37E+38  "
    LOCATE mm + 7, 3: PRINT " exponent range: -39 < E < +39 "
RETURN
```

```
' ******************** ConFactor conversion coefficients replacement routine ********************
ConFactorReplace:
    GOSUB SetupWindow2
    LOCATE mm + 5, 12
    PRINT "replace with conversion coefficients from channel #      "
ConFactorReplace1:
    GetInteger number, ESC, 3, mm + 5, 64, 0, 3
    IF ESC = 1 THEN '                          ESC pressed -- no replacement
      ELSEIF number < 0 OR number > 127 THEN
         LOCATE mm + 6, 16
         COLOR 30, 4: PRINT "  channel no. out of range        "
         BEEP
         GOTO ConFactorReplace1
      ELSE '                    (ESC=4) must be valid replacement
         FOR n = 0 TO 5: poly(ch, n) = poly(number, n): NEXT
    END IF
RETURN

' ************************************ Conversion Factor F-key ************************************
ConFactorKey:
    LOCATE 25, 2
    COLOR 12, 0: PRINT "F1"; : COLOR 11, 3: PRINT "T'couple";
    COLOR 12, 0: PRINT "F2"; : COLOR 9, 3: PRINT "T'mistor";
    COLOR 12, 0: PRINT "F3"; : COLOR 14, 3: PRINT "Linear";
    COLOR 12, 0: PRINT "F4"; : COLOR 10, 3: PRINT "Polynomial";
    COLOR 12, 0: PRINT "F8"; : COLOR 1, 3: PRINT "Remove";
    COLOR 12, 0: PRINT "PgUp"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(27); " "; CHR$(24); " "; CHR$(25); " "; CHR$(26); :
COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "PgDn"; : COLOR 1, 3: PRINT " ";
    COLOR 12, 0: PRINT "F9"; : COLOR 5, 3: PRINT "Done";
    COLOR 12, 0: PRINT "F10"; : COLOR 15, 3: PRINT "HELP";
RETURN
```

```
' ************************************** ConFactorHelp **************************************
ConFactorHelp:
   WindowOpen 8, 24 '          open HELP screen window
   GOTO ConFactorHelpPage1 '  display ConFactor HELP page 1

ConFactorHelp1:
   k$ = INKEY$

   SELECT CASE k$

      CASE CHR$(0) + CHR$(80), CHR$(0) + CHR$(81) '    ^_ or PgDn (Page Down)
         GOTO ConFactorHelpPage2

      CASE CHR$(0) + CHR$(72), CHR$(0) + CHR$(73) '    ^^ or PgUp (Page Up)
         GOTO ConFactorHelpPage1

      CASE CHR$(27), CHR$(0) + CHR$(68) '    Esc & F10 - leave HELP screen
         WindowClose 8, 24 '                close HELP screen window
         RETURN

      CASE ELSE
         GOTO ConFactorHelp1

   END SELECT


ConFactorHelpPage1: ' ********************** ConFactor HELP page 1 **********************
   LOCATE 8, 2: COLOR , 7: PRINT "  "; : COLOR 12, 0: PRINT "F1"; : COLOR 11, 7:
PRINT " - For thermocouple, a voltage is measured, and then converted to tempera-"
   LOCATE 9, 2: PRINT "       ture (°C).  The conversion coefficients of a Type T
(Copper-Constantan)"
   LOCATE 10, 2: PRINT "       thermocouple equation (from NBS Monograph 125) are
automatically in-   "
   LOCATE 11, 2: PRINT "       stalled, if no previously entered coefficients are available.
other   "
   LOCATE 12, 2: PRINT "       coefficient sets (e.g. for another thermocouple type) may
be entered.  "
   LOCATE 13, 2: PRINT "       A ± 5V input voltage range is also automatically
```

```
selected.                "
    LOCATE 14, 2: PRINT "  "; : COLOR 12  0: PRINT "F2"; : COLOR 9, 7: PRINT " - For
thermistor, a measured resistance (R) in Ω (Ohms) is converted to  "
    LOCATE 15, 2: PRINT "        temperature using  T = 1 / ["; : COLOR 1: PRINT "a"; :
COLOR 9: PRINT " + "; : COLOR 1: PRINT "b"; : COLOR 9: PRINT "(ln R) + "; : COLOR
1: PRINT "c"; : COLOR 9: PRINT "(ln R)^3] + "; : COLOR 1: PRINT "d"; : COLOR 9:
PRINT "  {Steinhart-  "
    LOCATE 16, 2: PRINT "        Hart equation}.  Coeffieients "; : COLOR 1: PRINT "a, b,
c "; : COLOR 9: PRINT "are determined (based on thermis-"
    LOCATE 17, 2: PRINT "        tor specification: 2252Ω @ 25°C) and automatically
installed.  "; : COLOR 1: PRINT "d"; : COLOR 9: PRINT " is an "
    LOCATE 18, 2: PRINT "        offset value (in °C).  If necessary, another set of
coefficients may be"
    LOCATE 19, 2: PRINT "        entered.  A ± 5V input voltage range is also
automatically selected.  "
    LOCATE 20, 2: PRINT "  "; : COLOR 12, 0: PRINT "F3"; : COLOR 14, 7: PRINT " -
This linear equation (multiplier "; : COLOR 6: PRINT "a"; : COLOR 14: PRINT " and offset
"; : COLOR 6: PRINT "b"; : COLOR 14: PRINT ") converts input data to"
    LOCATE 21, 2: PRINT "        recognizable units.  For example, a measured voltage (";
: COLOR 0: PRINT "x"; : COLOR 14: PRINT ") from an anemo-"
    LOCATE 22, 2: PRINT "        meter can be converted to air velocity (m/s) by:  "; :
COLOR 6: PRINT "a"; : COLOR 14: PRINT " * "; : COLOR 0: PRINT "x"; : COLOR 14:
PRINT " + "; : COLOR 6: PRINT "b"; : COLOR 14: PRINT " = m/s.      "
    LOCATE 23, 2: COLOR 0, 3: PRINT " ....MORE....  "; : COLOR 15: PRINT CHR$(25);
: COLOR 0: PRINT " or "; : COLOR 15: PRINT "PgDn"; : COLOR 0: PRINT " for next
screen    "; : COLOR 15: PRINT "Esc"; : COLOR 0: PRINT " or "; : COLOR 15: PRINT
"F10"; : COLOR 0: PRINT " to leave Help screen  "
    GOTO ConFactorHelp1


ConFactorHelpPage2: ' ********************** ConFactor HELP page 2 **********************
    LOCATE 8, 2: COLOR , 7: PRINT "  "; : COLOR 12, 0: PRINT "F4"; : COLOR 10, 7:
PRINT " - For some devices, e.g. thermal anemometers, a simple linear conversion "
    LOCATE 9, 2: PRINT "        may not be sufficient.  More elaborate conversion process
is necessary."
    LOCATE 10, 2: PRINT "        A 5th order polynomial equation: "; :
        COLOR 2: PRINT "A"; : COLOR 0: PRINT "x^5"; : COLOR 10: PRINT " + ";
        COLOR 2: PRINT "B"; : COLOR 0: PRINT "x^4"; : COLOR 10: PRINT " + ";
```

```
        COLOR 2: PRINT "C"; : COLOR 0: PRINT "x^3"; : COLOR 10: PRINT " + ";
        COLOR 2: PRINT "D"; : COLOR 0: PRINT "x^2"; : COLOR 10: PRINT " + ";
        COLOR 2: PRINT "E"; : COLOR 0: PRINT "x"; : COLOR 10: PRINT " + ";
        COLOR 2: PRINT "F    ": COLOR 10
    LOCATE 11, 2: PRINT "       is provided. "; : COLOR 2: PRINT "A, B, C, D, E, "; :
COLOR 10: PRINT "and "; : COLOR 2: PRINT "F"; : COLOR 10: PRINT " are the
polymonial coefficients, "; : COLOR 0: PRINT "x"; : COLOR 10: PRINT "  "
    LOCATE 12, 2: PRINT "       represents the measured voltage.
          "
    LOCATE 13, 2: PRINT "  "; : COLOR 12, 0: PRINT "F8"; : COLOR 1, 7: PRINT " -
Remove thermocouple/thermistor/linear/polynomial selection.          "
    LOCATE 14, 2: PRINT "       To change selection, it is not necessary to first remove
the old       "
    LOCATE 15, 2: PRINT "       selection.  Simply make new selection and enter the
new coefficients   "
    LOCATE 16, 2: PRINT "       (and overwrite the old selection and any old coefficients).
          "
    LOCATE 17, 2: PRINT "  "; : COLOR 12, 0: PRINT "F9"; : COLOR 5, 7: PRINT " -
When channel setup is completed, use F9 to exit to the main screen.    "
    LOCATE 18, 2: PRINT "       You will then have the opportunity to store this setup
screen, go to   "
    LOCATE 19, 2: PRINT "       another setup, start data acquisition, or save already
collected data. "
    LOCATE 20, 2: PRINT "  "; : COLOR 12, 0: PRINT "F10"; : COLOR 15, 7: PRINT " -
Whenever stuck, try F10 - HELP screen.                              "
    LOCATE 21, 2: PRINT "  "; : COLOR 12, 0: PRINT "Esc"; : COLOR 0, 7: PRINT " -
Always exit to the previous screen.  Here, Esc is equivalent to F9.   "
    LOCATE 22, 2: PRINT "  "; : COLOR 12, 0: PRINT CHR$(27); CHR$(18); CHR$(26); :
COLOR 4, 7: PRINT " - arrows change row and column.    "; : COLOR 12, 0: PRINT
"PgUp"; : COLOR 4, 7: PRINT " "; : COLOR 12, 0: PRINT "PgDn"; : COLOR 4, 7: PRINT
" - change display screen.   "
    LOCATE 23, 2: COLOR 0, 3: PRINT "              "; : COLOR 15: PRINT CHR$(24); :
COLOR 0: PRINT " or "; : COLOR 15: PRINT "PgUp"; : COLOR 0: PRINT " for previous
screen    "; : COLOR 15: PRINT "Esc"; : COLOR 0: PRINT " or "; : COLOR 15: PRINT
"F10"; : COLOR 0: PRINT " to leave Help screen  "
    GOTO ConFactorHelp1
```

```
' ************************************************************************************************
' ************************************** Unit label display ***************************************
' ************************************************************************************************
UnitLabel:
    SCREEN , , VS, VS
    COLOR 10, 0: LOCATE 4, 2: PRINT " Unit Label Selection "
    LOCATE 6, col: PRINT " Unit  "
    GOSUB NameUnitKey '                          UnitLabel F-key
    LOCATE 25, 10: COLOR 14: PRINT "Unit";
    LOCATE 25, 25: COLOR 1: PRINT "Unit";

UnitLabel1:
    LOCATE row, col: COLOR 16, 2: PRINT unit$(ch)
    rowt = row: colt = col: cht = ch
    NC = 0: ESC = 0

UnitLabel2:
    k$ = INKEY$

    SELECT CASE k$

        CASE CHR$(27), CHR$(0) + CHR$(67) '   Esc & F9 - Screen Setup Completed
            ESC = 1 '           exit subprogram, return to WhichFile
            NC = 1

        CASE CHR$(0) + CHR$(73) '    PgUp (Page Up)
            GOSUB PageUp

        CASE CHR$(0) + CHR$(81) '    PgDn (Page Down)
            GOSUB PageDown

        CASE CHR$(0) + CHR$(72) '    ↑ (Up Arrow)
            GOSUB UpArrow

        CASE CHR$(0) + CHR$(80) '    ↓ (Down Arrow)
            GOSUB DownArrow
```

```
        CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
            NC = 1
            IF colt = col5 THEN col = col4 ELSE col = col9

        CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
            NC = 1
            IF colt = col5 THEN
                col = col6: ch = ch + 16
              ELSE col = col1: ch = ch - 16
            END IF


        CASE CHR$(0) + CHR$(59) '    F1 - Enter unit label
            GOSUB SetupWindow1 '          (WindowOpen 7, 25)
            LOCATE mm + 2, 4: COLOR 0, 2
            PRINT " Enter unit label (7 characters max)          "
            temp$ = unit$(ch)
            GetLabel temp$, ESC, 7, mm + 2, 43, 0, 2 '  count,row,col,fclr,bclr
            IF ESC = 1 THEN '      ESC pressed, no label entered
              ELSEIF ESC = 4 THEN
                 GOSUB SetupWindow2
                 LOCATE mm + 5, 12
                 PRINT "replace with unit label from channel #     "
UnitLabel3:
                 GetInteger number, ESC, 3, mm + 5, 51, 0, 3
                 IF ESC = 1 THEN '            ESC pressed, no replacement
                   ELSEIF number < 0 OR number > 127 THEN
                      LOCATE mm + 6, 16
                      COLOR 30, 4: PRINT "  channel no. out of range       "
                      BEEP
                      GOTO UnitLabel3
                   ELSE unit$(ch) = unit$(number) ' must be valid replacement
                 END IF
              ELSE unit$(ch) = temp$
            END IF
            WindowClose 7, 25
```

```
        CASE CHR$(0) + CHR$(66) '    F8 - Remove unit label
            unit$(ch) = "none    "

        CASE CHR$(0) + CHR$(68) '    F10 - HELP Screen
            GOSUB UnitLabelHelp

        CASE ELSE
            GOTO UnitLabel2

    END SELECT


    LOCATE rowt, colt
    IF unit$(cht) = "none    " THEN
        COLOR , 7: PRINT SPC(7);
      ELSE COLOR 0: PRINT unit$(cht)
    END IF

    IF NC = 0 THEN '          new column/new screen?
        GOTO UnitLabel1 '           No
      ELSE ClearLine 4, 1, 4, 48, 7 '  Yes
        LOCATE 6, colt: COLOR 1: PRINT " Unit  "
    END IF

    IF ESC = 1 THEN GOTO SetupDone ELSE RETURN

' ****************************************** UnitLabel Help ******************************************
UnitLabelHelp:
    WindowOpen 8, 24 '   open HELP screen window
    LOCATE 8, 2: COLOR 0: PRINT " Unit Label Selection "; : COLOR , 7: PRINT " HELP
screen              "; : COLOR 15, 2: PRINT "Esc"; : COLOR 0, 7: PRINT " or "; : COLOR
15, 2: PRINT "F10"; : COLOR 0, 7: PRINT " to exit HELP screen "
    LOCATE 9, 2: PRINT "  "; : COLOR 12, 0: PRINT "F1"; : COLOR 14, 7: PRINT " - A
unit label can be entered for each channel.  For example, enter '°C' "
    LOCATE 10, 2: PRINT "        or 'deg. C' for a thermocouple, 'm/s' or 'FPM' for an
anemometer.  This"
    LOCATE 11, 2: PRINT "        will also be the unit label displayed on the data
acqusistion display  "
```

```
    LOCATE 12, 2: PRINT "       screen, and on the printer output.  Maximum 7
characters are allowed.  "
    LOCATE 13, 2: PRINT "       If necessary, all unit labels are filled with blanks to 7
char. long.  "
    GOSUB NameUnitHelp1
RETURN



' ********************************************************************************************************
' ****************************** Setup completed, Exit subprogram ******************************
' ********************************************************************************************************
SetupDone: '    This is the only place to exit the subprogram
    FOR mm = 0 TO 3 '               and return to main program
        SCREEN , , mm, VS
        COLOR 15, 1 '                   display REMINDER to save setup screen
        LOCATE 3, 2: PRINT " REMINDER: Is this setup screen saved? "
        LOCATE 4, 2: PRINT "            Use F5 key to store it       "
    NEXT
    SCREEN , , VS, VS
    ESC = 0
EXIT SUB
```

```
' ************************************************************************************************
' ******************************** Setup display input window frame ******************************
' ************************************************************************************************
SetupWindow1:
    WindowOpen 7, 25
    IF row > 15 THEN mm = 7: nn = 15 ELSE mm = 16: nn = 24
    COLOR 15, 7
    LOCATE mm, 2: PRINT
"╔═══════════════════════════════════════════════════════════════════════
╗"
    FOR m = mm + 1 TO nn - 1
        LOCATE m, 2: PRINT "‖
        ‖"
    NEXT
    LOCATE nn, 2: PRINT
"╚═══════════════════════════════════════════════════════════════════════
╝";
RETURN

' ************************************************************************************************
' ************************ Setup display input replacement window frame ***************************
' ************************************************************************************************
SetupWindow2:
    COLOR 5, 3
    LOCATE mm + 2, 10
        PRINT "╔═══════════════════════════════════════════════════════════╗"
    FOR m = mm + 3 TO nn - 2
    LOCATE m, 10: PRINT "‖                                                   ‖"
    NEXT
    LOCATE nn - 1, 10
        PRINT "╚═══════════════════════════════════════════════════════════╝";
    COLOR 0: LOCATE mm + 3, 12: PRINT "channel number range: 0 to 127"
RETURN
```

```
' ************************************************************************************************
' ***************************** Page Up (PgUp) key handling routine ******************************
' ************************************************************************************************
PageUp:
    NC = 1
    IF VS = 0 THEN '   when already on screen 0
        VS = 3 '     go to screen 3
        ch = ch + 96
      ELSE
        VS = VS - 1 '   else go to previous screen
        ch = ch - 32
    END IF
RETURN

' ************************************************************************************************
' ***************************** Page Down (PgDn) key handling routine ****************************
' ************************************************************************************************
PageDown:
    NC = 1
    IF VS = 3 THEN '   when already on screen 3
        VS = 0 '     go to screen 0
        ch = ch - 96
      ELSE
        VS = VS + 1 '   else advance to next screen
        ch = ch + 32
    END IF
RETURN

' ************************************************************************************************
' ******************************** Up Arrow key handling routine *********************************
' ************************************************************************************************
UpArrow:
    row = row - 1
    ch = ch - 1
    IF row < 8 THEN '   new column/new visual page
        NC = 1
        row = 23
        SELECT CASE col
```

```
            CASE col1
                col = col6: VS = VS - 1 '  go to previous screen
            CASE col2
                col = col7: VS = VS - 1
            CASE col3
                col = col8: VS = VS - 1
            CASE col4
                col = col9: VS = VS - 1
            CASE col5
                col = col10: VS = VS - 1
            CASE col6
                col = col1
            CASE col7
                col = col2
            CASE col8
                col = col3
            CASE col9
                col = col4
            CASE col10
                col = col5
        END SELECT
        IF ch < 0 THEN VS = 3: ch = 127 ' when retreated past 1st ch.
    END IF '                              go to screen 3 & channel 127
RETURN

' ************************************************************************************************
' ******************************* Down Arrow key handling routine ********************************
' ************************************************************************************************
DownArrow:
    row = row + 1
    ch = ch + 1
    IF row > 23 THEN '   new column/new visual page
        NC = 1
        row = 8
        SELECT CASE col
            CASE col1
                col = col6
            CASE col2
```

```
                col = col7
            CASE col3
                col = col8
            CASE col4
                col = col9
            CASE col5
                col = col10
            CASE col6
                col = col1: VS = VS + 1 '  advance to next screen
            CASE col7
                col = col2: VS = VS + 1
            CASE col8
                col = col3: VS = VS + 1
            CASE col9
                col = col4: VS = VS + 1
            CASE col10
                col = col5: VS = VS + 1
        END SELECT
        IF ch > 127 THEN VS = 0: ch = 0 ' when advanced past last ch.
    END IF '                            go to screen 0 & channel 0
RETURN

' ************************************************************************************************************
' ************************************************************************************************************

END SUB
```

```
' ##########################################################################
' ############################  AC-UTIL.BAS  ###############################
' ################  Utility subprograms of data acquisition program  ###############
' ##########################################################################

' Biophysics Data Acquisition System ac128.12  900628  Stephen KW. Chang

REM $DYNAMIC

COMMON SHARED /I/ ScrStore() AS INTEGER

DIM ScrStore(1999) AS INTEGER '              screen save/restore storage array
DIM ESC AS INTEGER '         Esc key pressed indicator flag (1=pressed, 0=no)

'          parameters passed to subprograms WindowOpen/WindowClose/ClearLine
DIM firstline AS INTEGER '  first line of screen to be saved/restored/cleared
DIM lastline AS INTEGER '   last line of screen to be saved/restored/cleared
' segment%                  segment address of screen storage array
' offset%                   offset address of screen storage array

'                    parameters passed to subprogram ClearLine
DIM firstcol AS INTEGER '   first column to be cleared
DIM lastcol AS INTEGER '    last column to be cleared

'  parameters passed to subprograms GetName, GetNumber, GetLabel & GetInteger
'   filename$ '          entered filename (may be filename or path name)
'   label$ '             entered label (may be device name or unit label)
DIM row AS INTEGER '      row number passed to subprograms
DIM col AS INTEGER '      column number passed to subprograms
DIM count AS INTEGER ' character/digit count of filename/number to be entered
DIM fcolor AS INTEGER '  display foreground color passed to subprograms
DIM bcolor AS INTEGER '  display background color passed to subprograms
DIM number AS SINGLE '   number entered

DIM m AS INTEGER, mm AS INTEGER '                dummy variables
DIM n AS INTEGER, nn AS INTEGER '                dummy variables
DIM k AS INTEGER, pp AS INTEGER, qq AS INTEGER ' dummy variables
```

```
REM $STATIC


' ########################################################################
' ########################  Clear a line or a rectangle ##################
' ########################################################################
' Subprogram to clear a line or a rectangle called by
'    passing (firstline, firstcol, lastline, lastcol, bcolor)
'       firstline = first line of screen to be cleared
'       firstcol = first column to be cleared
'       lastline = last line of screen to be cleared
'       lastcol = last column to be cleared
'       bcolor = background color used to clear the line(s)
'
SUB ClearLine (firstline AS INTEGER, firstcol AS INTEGER, lastline AS INTEGER, lastcol
AS INTEGER, bcolor AS INTEGER) STATIC

LOCATE , , 0: COLOR , bcolor '   turn off any blinking cursor
FOR n = firstline TO lastline
   LOCATE n, firstcol: PRINT " "; SPC(lastcol - firstcol);
NEXT

END SUB



' ########################################################################
' ####################  Get an integer number from keyboard  #############
' ########################################################################
' Subprogram to get an integer number from keyboard,
'    called by passing (integer number, ESC, digit count, row position,
'                       column position, foreground color, background color)
' called from: DataAcqPrinterSelect, TimeCycle, replace channel no.
'
SUB GetInteger (number, ESC AS INTEGER, count AS INTEGER, row AS INTEGER, col
AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) STATIC
```

```
LOCATE 25, 2
    COLOR 12, 0: PRINT CHR$(17); "-BackSpace"; : COLOR 1, 3: PRINT "delete last";
    COLOR 12, 0: PRINT "Delete"; : COLOR 11, 3: PRINT "current";
    COLOR 12, 0: PRINT " "; CHR$(27); CHR$(26); " "; : COLOR , 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(17); "┘"; : COLOR 15, 3: PRINT "enter ";
    COLOR 12, 0: PRINT "Insert"; : COLOR 14, 3: PRINT "/typeover mode";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "discard";

keyinp$ = MID$(STR$(number), 2) '   convert integer number to string keyinp$
'                                   also strip off sign byte
IF LEN(keyinp$) > count THEN ' If somehow old integer is too long, truncate it
    keyinp$ = LEFT$(keyinp$, count)
    n = count '                 count = no. of input digits needed
  ELSE n = LEN(keyinp$) 'n = dummy loop count for no. of input integer digits
END IF

LOCATE row, col, 1, 0, 8: COLOR bcolor + 8, fcolor: PRINT keyinp$;
ESC = 0 '    Esc key pressed indicator flag
nn = 1 '     nn=1 indicates we are at the first digit input
k = 1 '      k=0 insert mode, k=1 typeover mode

DO
    n = n + 1

GetInteger1:
    k$ = INKEY$

    SELECT CASE k$
        CASE ""
            GOTO GetInteger1

        CASE CHR$(27) '     Esc -- quit integer number entry subprogram
            ESC = 1 '           set Esc key flag = 1
            GOTO GetInteger3 '     discard number input

        CASE CHR$(13) '     Enter -- end of integer entry
            n = count
```

```
CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
    IF n > 1 THEN n = n - 2 ELSE n = 0 '  if n=first char. then n = 0

CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
    IF n >= count THEN
        n = count - 1
      ELSEIF n > LEN(keyinp$) THEN
        n = LEN(keyinp$)
    END IF

CASE CHR$(8) '      BackSpace -- remove last digit
    IF n > 1 THEN
        n = n - 2
        keyinp$ = LEFT$(keyinp$, n) + MID$(keyinp$, n + 2)
      ELSE n = 0 '                          if n=first char. then n = 0
    END IF

CASE CHR$(0) + CHR$(83) '                Delete -- delete current digit
    keyinp$ = LEFT$(keyinp$, n - 1) + MID$(keyinp$, n + 1)
    n = n - 1

CASE CHR$(0) + CHR$(82) '       Insert -- insert/typeover mode switch
    IF n >= count THEN n = count - 1 ELSE n = n - 1
    IF k = 0 THEN
        k = 1: LOCATE , , 1, 0, 8 '   typeover mode
      ELSE k = 0: LOCATE , , 1, 6, 8 '   insert mode
    END IF

CASE "0", "1", "2", "3", "4", "5", "6", "7", "8", "9" '  valid digits
    IF nn = 1 THEN '      if on first input, a new digit is entered
        keyinp$ = k$ '  keep it, and delete previous integer
        n = 1 '         adjust loop count = 1
      ELSE '                          Else concatenate integer digits
        IF k = 1 THEN ' typeover mode
            keyinp$ = LEFT$(keyinp$, n - 1) + k$ + MID$(keyinp$, n + 1)
          ELSE '  (k=0) insert mode
            keyinp$ = LEFT$(keyinp$, n - 1) + k$ + MID$(keyinp$, n)
            keyinp$ = LEFT$(keyinp$, count) 'if number too long, truncate
```

```
            END IF
        END IF

    CASE IS < CHR$(32) '    disallow non-printable characters
        n = n - 1: BEEP
        LOCATE row + 1, 16, 0
        COLOR 30, 4: PRINT " Non-printable character entered "
        IF n >= count THEN n = count - 1
        GOTO GetInteger2

    CASE ELSE '          must be invalid input
        n = n - 1: BEEP
        LOCATE row + 1, 16, 0
        COLOR 30, 4: PRINT "  Invalid number input   ";
        COLOR 31, 4: PRINT k$; "       "
        IF n >= count THEN n = count - 1
        GOTO GetInteger2
  END SELECT

  LOCATE row + 1, 16, 0: COLOR , bcolor: PRINT SPC(33); 'remove any leftover
GetInteger2: '                                          warning message
  COLOR fcolor, bcolor: LOCATE row, col, 0: PRINT SPC(count);
                        LOCATE row, col: PRINT keyinp$;
                        LOCATE row, col, 1: PRINT LEFT$(keyinp$, n);

nn = 0 '   nn=0 means we have past the first digit input
LOOP UNTIL n = count


number = VAL(keyinp$) '   convert string keyinp$ back to numeric number

GetInteger3:
LOCATE , , 0: COLOR fcolor, bcolor ' turns off blinking cursor, restore color
END SUB
```

```
' ##########################################################################
' ########################  Input a label from keyboard  ####################
' ##########################################################################
' Subprogram to input a label (name or unit label) from keyboard,
'    called by passing (label, ESC, character count, row position,
'                       column position, foreground color, background color)
' called from: DeviceName, UnitLabel
'

SUB GetLabel (label$, ESC AS INTEGER, count AS INTEGER, row AS INTEGER, col AS
INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) STATIC

LOCATE 25, 2
    COLOR 12, 0: PRINT CHR$(17); "-BkSpace"; : COLOR 1, 3: PRINT "delete last";
    COLOR 12, 0: PRINT "Delete"; : COLOR 11, 3: PRINT "current";
    COLOR 12, 0: PRINT CHR$(27); CHR$(26); : COLOR , 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(17); "┘"; : COLOR 15, 3: PRINT "enter ";
    COLOR 12, 0: PRINT "F5"; : COLOR 10, 3: PRINT "replace";
    COLOR 12, 0: PRINT "Insert"; : COLOR 14, 3: PRINT "/typeover";
COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "discard";

IF label$ = "none    " THEN
    label$ = ""
    n = 0 '                         n = dummy loop count for input characters
  ELSEIF LEN(label$) > count THEN
    label$ = LEFT$(label$, count) 'if somehow label is too long, truncate it
    n = count '     count = no. of input characters needed
  ELSE n = count '     previous label exists
END IF

IF count > 7 THEN n = LEN(label$) '  this takes care of study ID remark entry

LOCATE row, col, 1, 0, 8: COLOR bcolor, fcolor: PRINT label$;
ESC = 0 '   Esc key pressed indicator flag
k = 1 '      k=0 insert mode, k=1 typeover mode

DO '            loop until n=count
    n = n + 1
```

```
GetLabel1:
   k$ = INKEY$

   SELECT CASE k$
      CASE ""
         GOTO GetLabel1

      CASE CHR$(27) '     Esc -- quit label entry subprogram
         ESC = 1 '          set Esc key flag = 1
         GOTO GetLabel4 '       discard label input

      CASE CHR$(13) '     Enter -- end of label entry
         n = count

      CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
         IF n > 1 THEN n = n - 2 ELSE n = 0 '  if n=first char. then n = 0

      CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
         IF n >= count THEN
            n = count - 1
          ELSEIF n > LEN(label$) THEN
            n = LEN(label$)
         END IF

      CASE CHR$(8) '      BackSpace -- delete previous character
         IF n > 1 THEN
            n = n - 2
            label$ = LEFT$(label$, n) + MID$(label$, n + 2)
          ELSE n = 0 '                        if n=first char. then n = 0
         END IF

      CASE CHR$(0) + CHR$(83) '          Delete -- delete current character
         label$ = LEFT$(label$, n - 1) + MID$(label$, n + 1)
         n = n - 1
```

```
        CASE CHR$(0) + CHR$(82) '       Insert -- insert/typeover mode switch
            IF n >= count THEN n = count - 1 ELSE n = n - 1
            IF k = 0 THEN
                k = 1: LOCATE , , 1, 0, 8 '   typeover mode
              ELSE k = 0: LOCATE , , 1, 6, 8 '   insert mode
            END IF

        CASE CHR$(0) + CHR$(63) '   F5 -- replace current label with label of
            ESC = 4 '       another channel (set indicator flag: ESC=4)
            GOTO GetLabel4

        CASE IS < CHR$(32) '    disallow non-printable characters
            n = n - 1: BEEP
            LOCATE row + 2, 8, 0: COLOR 30, 4
            PRINT "  Non-printable character entered  "
            IF n >= count THEN n = count - 1
            GOTO GetLabel2

        CASE ELSE '             must be valid character
            IF n > count THEN '   if on first input, a new character is
                label$ = k$ '   entered, keep it, and delete previous label
                n = 1 '         adjust loop count = 1
              ELSE '                                  Else concatenate label
                IF k = 1 THEN ' typeover mode
                        label$ = LEFT$(label$, n - 1) + k$ + MID$(label$, n + 1)
                  ELSE '  (k=0) insert mode
                    label$ = LEFT$(label$, n - 1) + k$ + MID$(label$, n)
                    label$ = LEFT$(label$, count) 'if label too long, truncate
                END IF
            END IF
    END SELECT

    LOCATE row + 2, 8, 0: COLOR , 7: PRINT SPC(35); '   remove any leftover
GetLabel2: '                                        warning message
    COLOR fcolor, bcolor: LOCATE row, col, 0: PRINT SPC(count);
                          LOCATE row, col: PRINT label$;
                          LOCATE row, col, 1: PRINT LEFT$(label$, n);
```

```
LOOP UNTIL n = count

IF count > 7 GOTO GetLabel4 '   if study ID entry, then don't pad with blanks

IF label$ = "" THEN label$ = "none    ": GOTO GetLabel4 '  no label entered
FOR m = 1 TO LEN(label$) '                              check if label is blank
    IF MID$(label$, m, 1) <> " " GOTO GetLabel3 '   i.e. all SPACE char.
NEXT
label$ = "none    " '   entered label must be a blank
GOTO GetLabel4

GetLabel3: '                              pad label$ up to count (7) char. long
IF LEN(label$) < count THEN '               to ensure a neat printer output
    FOR m = LEN(label$) TO count - 1 '  need -1 here to get 7 char.
        label$ = label$ + " "  '          otherwise gets 8 char.
    NEXT
END IF

GetLabel4:
LOCATE , , 0: COLOR fcolor, bcolor ' turns off blinking cursor, restore color
END SUB
```

```
' ##########################################################################
' ######################  Input a filename from keyboard  ###################
' ##########################################################################
' Subprogram to input a filename from keyboard,
'    called by passing (filename, ESC, character count, row position,
'                       column position, foreground color, background color)
' called from: LoadSetup, SaveSetup, SaveData, DataAcqFilename, ID remark
'
SUB GetName (filename$, ESC AS INTEGER, count AS INTEGER, row AS INTEGER, col
AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) STATIC

LOCATE 25, 2
    COLOR 12, 0: PRINT CHR$(17); "-BackSpace"; : COLOR 1, 3: PRINT "delete last";
    COLOR 12, 0: PRINT "Delete"; : COLOR 11, 3: PRINT "current";
    COLOR 12, 0: PRINT " "; CHR$(27); CHR$(26); " "; : COLOR , 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(17); "┘"; : COLOR 15, 3: PRINT "enter ";
    COLOR 12, 0: PRINT "Insert"; : COLOR 14, 3: PRINT "/typeover mode";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "discard";

IF LEN(filename$) > count THEN ' if somehow filename is too long, truncate it
    filename$ = LEFT$(filename$, count)
    n = count '             count = no. of input characters needed
  ELSE n = LEN(filename$) '   previous filename exists
END IF

LOCATE row, col: COLOR fcolor, bcolor: PRINT SPC(LEN(filename$) + 1);
LOCATE row, col, 1, 6, 8: COLOR 7, 1: PRINT filename$;
ESC = 0 '   Esc key pressed indicator flag
nn = 1 '    nn is flag (nn=1) to indicate the first character input
k = 0 '     k=0 insert mode, k=1 typeover mode


DO '            loop until n=count
    n = n + 1

GetName1:
    k$ = INKEY$
```

```
SELECT CASE k$
    CASE ""
        GOTO GetName1

    CASE CHR$(27) '    Esc -- quit filename entry subprogram
        ESC = 1 '          set Esc key flag = 1
        GOTO GetName4 '        discard filename input

    CASE CHR$(13) '    Enter -- end of filename entry
        LOCATE row, col: PRINT filename$;
        GOTO GetName3

    CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
        IF n > 1 THEN n = n - 2 ELSE n = 0 '  if n=first char. then n = 0

    CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
        IF n >= count THEN
            n = count - 1
          ELSEIF n > LEN(filename$) THEN
            n = LEN(filename$)
        END IF

    CASE CHR$(8) '     BackSpace -- remove previous character
        IF n > 1 THEN
            n = n - 2
            filename$ = LEFT$(filename$, n) + MID$(filename$, n + 2)
          ELSE n = 0 '                          if n=first char. then n = 0
        END IF

    CASE CHR$(0) + CHR$(83) '          Delete -- delete current character
        filename$ = LEFT$(filename$, n - 1) + MID$(filename$, n + 1)
        n = n - 1

    CASE CHR$(0) + CHR$(82) '        Insert -- insert/typeover mode switch
        IF n >= count THEN n = count - 1 ELSE n = n - 1
        IF k = 0 THEN
            k = 1: LOCATE , , 1, 0, 8 '   typeover mode
          ELSE k = 0: LOCATE , , 1, 6, 8 '   insert mode
```

```
        END IF

    CASE IS < CHR$(32) '    disallow non-printable characters
        n = n - 1: BEEP
        LOCATE row + 1, col, 0
        COLOR 30, 4: PRINT " Non-printable character entered "
        IF n >= count THEN n = count - 1
        GOTO GetName2
                                        invalid filename characters
    CASE CHR$(34), "/", "[", "]", "|", "<", ">", "+", "=", ";", ","
        BEEP
        n = n - 1
        LOCATE row + 1, col, 0
        COLOR 30, 4: PRINT " Invalid filename character  ";
        COLOR 31, 4: PRINT k$; "   "
        IF n >= count THEN n = count - 1
        GOTO GetName2

    CASE ELSE '           must be valid character
        IF nn = 1 THEN '    if on first input, a new character is entered
            filename$ = k$ '   keep it, and delete previous filename
            n = 1 '            adjust loop count = 1
          ELSE '                                Else concatenate filename
            IF k = 1 THEN ' typeover mode
                filename$ = LEFT$(filename$, n - 1) + k$ + MID$(filename$, n + 1)
              ELSE '  (k=0) insert mode
                filename$ = LEFT$(filename$, n - 1) + k$ + MID$(filename$, n)
                filename$ = LEFT$(filename$, count)'if filename too long, truncate
            END IF
        END IF

END SELECT
```

```
    LOCATE row + 1, col, 0: COLOR , bcolor: PRINT SPC(33); '    remove any
GetName2: '                                                    leftover warning message
    COLOR fcolor, bcolor: LOCATE row, col, 0: PRINT SPC(count);
                          LOCATE row, col: PRINT filename$;
                          LOCATE row, col, 1: PRINT LEFT$(filename$, n);

nn = 0 '     nn=0 means we are past the first character input
LOOP UNTIL n = count


GetName3:
IF filename$ = "" THEN '     Check for null string filename
    LOCATE row + 1, col, 0
    COLOR 30, 4: PRINT " No filename                    "
    BEEP: GOTO GetName2
END IF
FOR m = 1 TO LEN(filename$) '                  check if the file extension
    IF MID$(filename$, m, 1) = "." THEN  '   indicator "." was entered
        IF LEN(filename$) - m > 3 THEN '      allow only 3 more characters
            LOCATE row + 1, col, 0 '         for the extension
            COLOR 30, 4: PRINT " file extension too long          "
            BEEP: GOTO GetName2
          ELSEIF LEN(filename$) - m = 0 THEN '  no character was entered
            LOCATE row + 1, col, 0 '           for the extension
            COLOR 30, 4: PRINT " no file extension after";
            COLOR 31, 4: PRINT "  '. '   "
            BEEP: GOTO GetName2
        END IF
    END IF
NEXT

GetName4:
LOCATE , , 0: COLOR fcolor, bcolor ' turns off blinking cursor, restore color
END SUB
```

```
' ##############################################################################
' ######################  Input a number from keyboard  ########################
' ##############################################################################
' Subprogram to input a number from keyboard,
'    called by passing (number, ESC, digit count, row position,
'                       column position, foreground color, background color)
' called from: ConFactor, PlotData
'
SUB GetNumber (number AS SINGLE, ESC AS INTEGER, count AS INTEGER, row AS
INTEGER, col AS INTEGER, fcolor AS INTEGER, bcolor AS INTEGER) STATIC

LOCATE 25, 2
    COLOR 12, 0: PRINT CHR$(17); "-"; : COLOR 1, 3: PRINT "delete last";
    COLOR 12, 0: PRINT "Delete"; : COLOR 9, 3: PRINT "current";
    COLOR 12, 0: PRINT CHR$(27); CHR$(25); CHR$(26); : COLOR , 3: PRINT " ";
    COLOR 12, 0: PRINT CHR$(17); "┘"; : COLOR 15, 3: PRINT "enter ";
    COLOR 12, 0: PRINT "F5"; : COLOR 10, 3: PRINT "replace";
    COLOR 12, 0: PRINT "F9"; : COLOR 11, 3: PRINT "keep";
    COLOR 12, 0: PRINT "Insert"; : COLOR 14, 3: PRINT "/typeover";
    COLOR 12, 0: PRINT "Esc"; : COLOR 0, 3: PRINT "discard";

IF row > 15 THEN pp = 16 ELSE pp = 7
keyinp$ = STR$(number) '        convert number to string keyinp$
IF LEN(keyinp$) > count THEN ' If somehow old number is too long, truncate it
    keyinp$ = LEFT$(keyinp$, count)
    n = count '         count = no. of input digits needed
  ELSE n = LEN(keyinp$) ' n = dummy loop count for no. of input number digits
END IF

LOCATE row, col, 1, 6, 8: COLOR fcolor, bcolor: PRINT keyinp$;
ESC = 0 '   Esc key pressed indicator flag
nn = 1 '    nn=1 flag -- indicates it's the first input digit
qq = 0 '    qq flag -- exponent designator position
k = 0 '     k=0 insert mode, k=1 typeover mode

DO '            loop until n=count
    n = n + 1
```

```
GetNumber1:
   k$ = INKEY$

   SELECT CASE k$

      CASE ""
         GOTO GetNumber1

      CASE CHR$(27) '      Esc -- quit number entry subprogram
         ESC = 1 '           set Esc key flag = 1
         GOTO GetNumber5 '       discard number input

      CASE CHR$(13), CHR$(0) + CHR$(80) '    Enter/^_ -- end of number entry
         COLOR 15, bcolor: LOCATE row, col: PRINT keyinp$;
         GOTO GetNumber4

      CASE CHR$(0) + CHR$(75) '    ← (Left Arrow)
         IF n > 1 THEN n = n - 2 ELSE n = 0 '  if n=first char. then n = 0

      CASE CHR$(0) + CHR$(77) '    → (Right Arrow)
         IF n >= count THEN
            n = count - 1
          ELSEIF n > LEN(keyinp$) THEN
            n = LEN(keyinp$)
         END IF

      CASE CHR$(8) '      BackSpace -- remove previous digit
         IF n > 1 THEN
            n = n - 2
            keyinp$ = LEFT$(keyinp$, n) + MID$(keyinp$, n + 2)
          ELSE n = 0 '                          if n=first char. then n = 0
         END IF
         IF qq > n THEN qq = 0 ' if deleted exp. desig. then updata qq flag

      CASE CHR$(0) + CHR$(83) '              Delete -- delete current digit
         keyinp$ = LEFT$(keyinp$, n - 1) + MID$(keyinp$, n + 1)
         n = n - 1
```

```
        CASE CHR$(0) + CHR$(82) '       Insert -- insert/typeover mode switch
            IF n >= count THEN n = count - 1 ELSE n = n - 1
            IF k = 0 THEN
                k = 1: LOCATE , , 1, 0, 8 '   typeover mode
              ELSE k = 0: LOCATE , , 1, 6, 8 '   insert mode
            END IF

        CASE CHR$(0) + CHR$(63) '   F5 -- replace current label with label of
            ESC = 4 '       another channel (set indicator flag: ESC=4)
            GOTO GetNumber5

        CASE CHR$(0) + CHR$(67) '   F9 - keep the rest of the numbers
            ESC = 2 '               set ESC = 2 as the F1 key indicator
            GOTO GetNumber5 '       keep numbers entered thus far

        CASE "0", "1", "2", "3", "4", "5", "6", "7", "8", "9", "+", "-", "."
            GOTO GetNumber2 '               valid digits, signs & decimal point

        CASE "E", "e", "!", "D", "d", "#" '       valid exponent designations
            IF qq = 0 THEN '        qq<>0 flag -- exp. desig. already entered
                qq = n
              ELSE '                can not accept more than 1 exp. designator
                n = n - 1: BEEP
                LOCATE pp + 3, 3, 0
                COLOR 30, 4: PRINT " Duplicate exponent designator   ";
                COLOR 31, 4: PRINT k$; "  "
                GOTO GetNumber3
            END IF
GetNumber2:
            IF nn = 1 THEN '      if on first input, a new digit is entered
                keyinp$ = k$ '  keep it, and delete previous number
                n = 1 '         adjust loop count = 1
              ELSE '                            Else concatenate number digits
                IF k = 1 THEN ' typeover mode
                    keyinp$ = LEFT$(keyinp$, n - 1) + k$ + MID$(keyinp$, n + 1)
                  ELSE '  (k=0) insert mode
                    keyinp$ = LEFT$(keyinp$, n - 1) + k$ + MID$(keyinp$, n)
                    keyinp$ = LEFT$(keyinp$, count) 'if number too long, truncate
```

```
            END IF
        END IF

    CASE IS < CHR$(32) '    disallow non-printable characters
        n = n - 1: BEEP
        LOCATE pp + 3, 3, 0
        COLOR 30, 4: PRINT " Non-printable character entered    "
        IF n >= count THEN n = count - 1
        GOTO GetNumber3

    CASE ELSE '          must be invalid input
        n = n - 1: BEEP
        LOCATE pp + 3, 3, 0
        COLOR 30, 4: PRINT " Invalid number input   ";
        COLOR 31, 4: PRINT k$; "            "
        IF n >= count THEN n = count - 1
        GOTO GetNumber3

  END SELECT

  LOCATE pp + 3, 3, 0: COLOR , 7: PRINT SPC(36); ' remove any leftover
GetNumber3: '                                      warning message
  COLOR 15, bcolor: LOCATE row, col, 0: PRINT SPC(count);
                    LOCATE row, col: PRINT keyinp$;
                    LOCATE row, col, 1: PRINT LEFT$(keyinp$, n);

nn = 0 '     nn=0 -- we have past the first input digit
LOOP UNTIL n = count
```

```
GetNumber4: '          check for input number range to avoid overflow error
mm = LEN(keyinp$) '   get length of keyinp$
FOR m = 1 TO mm '                    check if exponent (E,e,D,d) was entered
    IF MID$(keyinp$, m, 1) > "9" THEN
        nn = VAL(MID$(keyinp$, m + 1, mm)) '  convert exponent to numeric no.
        IF nn < -38 OR nn > 38 THEN '      disallow exponent > sigle precision
            BEEP '                          numbers (i.e. - 38 <= E <= 38)
            LOCATE pp + 3, 3, 0: COLOR 30, 4
            IF nn > 38 THEN
                PRINT " Exponent          too large          "
              ELSE PRINT " Exponent          too small          "
            END IF
            LOCATE pp + 3, 15: COLOR 31: PRINT nn
            IF n >= count THEN n = count - 1
            GOTO GetNumber3
        END IF
    END IF
NEXT

'    convert string keyinp$ to double precision numeric number, number may be
nn# = VAL(keyinp$) '        too large to cause single precision overflow error
IF nn# < -3.37E+38 OR nn# > 3.37E+38 THEN
    BEEP '                          when exponent is OK, mantissa
    LOCATE pp + 3, 3, 0 '             mantissa may still be out of range
    COLOR 30, 4: PRINT "                 out of range      "
    LOCATE pp + 3, 4: COLOR 31: PRINT nn#
    IF n >= count THEN n = count - 1
    GOTO GetNumber3
  ELSE number = nn# '   number OK, convert to single precision
END IF

GetNumber5:
LOCATE , , 0: COLOR 15, bcolor '   turns off blinking cursor, restore color
END SUB
```

```
' ************************************************************************************************
' ***************** Close Message Window - by restoring the screen content *****************
' ************************************************************************************************
' Subprogram to close a message window.
'    by restoring screen content from firstline to lastline
'
SUB WindowClose (firstline AS INTEGER, lastline AS INTEGER) STATIC
    segment% = VARSEG(ScrStore(0))
    offset% = VARPTR(ScrStore(0))
'      segment% & offset% point to location of stored screen content
'      firstline & lastline give position on the display screen
    CALL ScrRest(firstline, lastline, segment%, offset%)
END SUB

' ************************************************************************************************
' ******************** Open message window - by saving the screen content ********************
' ************************************************************************************************
' Subprogram to open a message window,
'    by saving screen content from firstline% to lastline%
'
SUB WindowOpen (firstline AS INTEGER, lastline AS INTEGER) STATIC
    segment% = VARSEG(ScrStore(0))
    offset% = VARPTR(ScrStore(0))
'      segment% & offset% point to location of stored screen content
'      firstline & lastline give position on the display screen
    CALL ScrSave(firstline, lastline, segment%, offset%)
END SUB
```

```
;-------------------------------------------------------------------------------------------------------------|
; Subroutine  Acq  acquires data from the A/D board  (base address = 330h)                                    |
;  Acq is called from main program as  Call Acq(comcode%, digout%, rad%)                                      |
;                                                    comcode% = channel select + gain                       |
;                                                    digout% = digital output (inverted)                    |
;                                                    rad% = raw A/D input data                              |
;-------------------------------------------------------------------------------------------------------------|
Code    SEGMENT   byte      public     'code'
        ASSUME    cs:Code
        PUBLIC    Acq

Acq     PROC      FAR       ; entry point for subroutine

        push  BP            ; standard procedure, store base pointer
        mov   BP,SP         ; standard procedure, get stack pointer

        mov   DX,339H       ; set flag clear byte address
        out   DX,AL         ; clear flags

        mov   DX,331H       ; gain/channel mux select register address
        mov   BX,[BP]+10    ; get address of Gain/Select
        mov   AL,[BX]       ; get Gain code/Channel mux select
        out   DX,AL         ; set gain & channel mux

        mov   DX,33BH       ; digital output byte register address
        mov   BX,[BP]+8     ; get digital output, it has the
        mov   AL,[BX]       ;   multiplexed sub-channel no.
        out   DX,AL         ; set digital output byte (inverted)

        mov   DX,332H       ; set start A/D conversion register address
        out   DX,AL         ; writing to it starts the A/D conversion

        mov   DX,330H       ; set status register address
                            ; loop, waiting for A/D to finish
Loop:   in    AL,DX         ; read status register
        and   AL,40H        ; check Done bit
        jz    Loop          ; loop if A/D not done
                            ; DONE, get A/D data
```

```
        mov     DX,333H         ; set A/D low byte register address
        in      AX,DX           ; read in A/D data
                                ; automatically clear Done bit
        mov     BX,[BP]+6       ; get rad (A/D input data) address in stack
        mov     [BX],AX         ; store A/D data in stack

        pop     BP              ; restore base pointer
        ret     6               ; restore stack, and return to main program
Acq     ENDP
Code    ENDS
        END
```

```
;-------------------------------------------------------------------------------------------------------|
; Subroutine  PrStat  reads the printer status byte to                                                  |
;                                  detect I/O error and out-of-paper conditions                         |
; PrStat is called from main program as                                                                 |
;          Call PrStat(printerNo, flagPS)                                                               |
;                          printerNo = printer number (0, 1, or 2)                                      |
;                          flagPS = printer status byte: bit 3 = printer I/O error                      |
;                                                        bit 5 = out-of-paper error                     |
;-------------------------------------------------------------------------------------------------------|
Code     SEGMENT   byte      public     'code'
         ASSUME    cs:Code
         PUBLIC    PrStat

PrStat   PROC      FAR       ; entry point for subroutine

         push  BP            ; standard procedure, store base pointer
         mov   BP,SP         ; standard procedure, get stack pointer

         mov   BX,[BP]+8     ; get address of printer no.
         mov   DX,[BX]       ; printer no. in DX
         mov   AH,2          ; AH=2 -- read printer status
         int   17h           ; printer I/O interrupt
                             ; status comes back in AH
         and   AH,28h        ; zero out unwanted bits, only want bits 3 & 5
                             ; bit 3 = I/O error
                             ; bit 5 = out-of-paper
         mov   AL,AH         ; put status into AL
         xor   AH,AH         ; zero out AH
         mov   BX,[BP]+6     ; get flagPS address in stack
         mov   [BX],AX       ; store printer status byte in flagPS

         pop   BP            ; restore base pointer
         ret   4             ; restore stack, and return to main program
PrStat   ENDP
Code     ENDS
         END
```

```
;-------------------------------------------------------------------------------------------------------------|
; Subroutine  ScrSave  saves the screen for later restoration,                                                |
;                                  in preparation to create a task window                                     |
; ScrSave is called from main program as                                                                      |
;          Call ScrSave(firstline, lastline, segment, offset)                                                 |
;                              firstline, lastline = first and last line of screen to be saved                |
;                              segment, offset = segment and offset of the storage array                      |
;                                                                                                             |
;          Movement of data is   DS:[SI] --> ES:[DI]                                                          |
;-------------------------------------------------------------------------------------------------------------|
Code     SEGMENT   Byte     Public    'Code'
         ASSUME    CS:Code
         PUBLIC    ScrSave

ScrSave  PROC      FAR      ;entry point for subroutine

         push  BP           ;save base pointer
         mov   BP,SP        ;get stack pointer
         push  DS           ;save data segment register
         push  ES           ;save extra segment register

         mov   SI,[BP+6]    ;get address of offset of first array element
         mov   DI,[SI]      ;address of first element in storage array
         mov   SI,[BP+8]    ;get address of segment of storage array
         mov   ES,[SI]      ;ES has segment of storage array

         mov   SI,[BP+10]   ;get address of LASTLINE
         mov   AL,[SI]      ;LASTLINE into AL
         mov   CL,160       ;prepare to multiply by 160 (2 * 80 word/line)
         mul   CL           ;AL * CL = AX (result in AX)
         mov   BX,AX        ;BX has last screen address to save

         mov   SI,[BP+12]   ;get address of FIRSTLINE
         mov   AL,[SI]      ;FIRSTLINE into AL
         dec   AL           ;adjust 1-25 to 0-24 line (screen line no.)
         mul   CL           ;AL * CL = AX (result in AX)
         mov   SI,AX        ;SI points to starting address on screen
```

```
            sub     BX,AX           ;calculate no. of bytes to copy
            mov     CX,BX           ;CX has no. of words to copy
            shr     CX,1            ;divide CX by 2 (shift right) for no. of words
                                    ;CX has loop count to be used for MOVSW inst.
            mov     AH,0Fh          ;setup to get current video page no.
            int     10h             ;current page no. returned in BH
            mov     AL,BH           ;page no. into AL
            xor     AH,AH           ;zero out AH, AX now has current page no.
            mov     BX,0B800h       ;starting address for EGA (video page 0)
            cmp     AX,0            ;is this video page 0?
            jz      Continue

AddPage:    add     BX,100h         ;not page 0, thus adjust segment to next page
            dec     AX              ;decrement page no. by one
            jz      Continue        ;is it video page 0 now?
            jmp     AddPage         ;no, go increment page

Continue:   mov     DS,BX           ;yes, set DS to (EGA) screen segment

            cld                     ;direction flag = forward
            rep     movsw           ;move data  DS:[SI] -> ES:[DI]
                                    ;loop count in CX
            pop     ES              ;restore extra segment register
            pop     DS              ;restore data segment register
            pop     BP              ;restore base pointer
            ret     8               ;return to main program
                                    ;skip stack locations of the passed parameters
ScrSave     ENDP
Code        ENDS
            END
```

```
;-------------------------------------------------------------------------------------------------------|
; Subroutine  ScrRest  restores the screen to a state                                                   |
;                                  prior to creation of the task window                                 |
; ScrRest is called from main program as                                                                |
;          Call ScrRest(firstline, lastline, segment, offset)                                           |
;                        firstline, lastline = first and last line of screen to be saved                |
;                        segment, offset = segment and offset of the storage array                      |
;                                                                                                       |
;          Movement of data is   DS:[SI] --> ES:[DI]                                                    |
;-------------------------------------------------------------------------------------------------------|
Code      SEGMENT   Byte      Public     'Code'
          ASSUME    CS:Code
          PUBLIC    ScrRest

ScrRest   PROC      FAR       ;entry point for subroutine

          push  BP            ;save base pointer
          mov   BP,SP         ;get stack pointer
          push  DS            ;save data segment register
          push  ES            ;save extra segment register

          mov   AH,0Fh        ;setup to get current video page no.
          int   10h           ;current page no. returned in BH
          mov   AL,BH         ;page no. into AL
          xor   AH,AH         ;zero out AH, AX has current video page no.
          mov   BX,0B800h     ;starting address for EGA (video page 0)
          cmp   AX,0          ;is this video page 0?
          jz    Continue

AddPage:  add   BX,100h       ;not page 0, thus adjust segment to next page
          dec   AX            ;decrement page no. by one
          jz    Continue      ;is it video page 0 now?
          jmp   AddPage       ;no, go increment page

Continue: mov   ES,BX         ;yes, set ES to (EGA) screen segment

          mov   DI,[BP+10]    ;get address of LASTLINE
          mov   AL,[DI]       ;LASTLINE into AL
```

```
        mov     CL,160          ;prepare to multiply by 160 (2 * 80 word/line)
        mul     CL              ;AL * CL = AX (result in AX)
        mov     BX,AX           ;BX has last screen address to restore

        mov     DI,[BP+12]      ;get address of FIRSTLINE
        mov     AL,[DI]         ;FIRSTLINE into AL
        dec     AL              ;adjust 1-25 to 0-24 line (screen line no.)
        mul     CL              ;AL * CL = AX (result in AX)
        mov     DI,AX           ;DI points to starting address on screen

        sub     BX,AX           ;calculate no. of bytes to copy
        mov     CX,BX           ;CX has no. of words to copy
        shr     CX,1            ;divide CX by 2 (shift right) for no. of words
                                ;CX has loop count to be used for MOVSW inst.
        mov     SI,[BP+8]       ;get address of segment of storage array
        mov     AX,[SI]         ;segment of storage array in AX
        mov     SI,[BP+6]       ;get address of offset of first array element
        mov     SI,[SI]         ;address of first element in storage array
        mov     DS,AX           ;DS now gets segment of storage array

        cld                     ;direction flag = forward
        rep     movsw           ;move data  DS:[SI] -> ES:[DI]
                                ;loop count in CX
        pop     ES              ;restore extra segment register
        pop     DS              ;restore data segment register
        pop     BP              ;restore base pointer
        ret     8               ;return to main program
                                ;skip stack locations of the passed parameters
ScrRest ENDP
Code    ENDS
        END
```

DISTRIBUTION LIST

10 Copies to:

Commandant
Academy of Health Sciences
ATTN: HSHA-FR (USAMRDC Liaison Officer)
Fort Sam Houston, TX 78234-6100

4 Copies to:

Defense Technical Information Center
ATTN: DTIC-SDAC
Alexandria, VA 22304-6145

2 Copies to:

Commander
U.S. Army Medical Research and Development Command
ATTN: SGRD-OP
Fort Detrick
Frederick, MD 21702-5012

Commander
U.S. Army Medical Research and Development Command
ATTN: SGRD-PLE
Fort Detrick
Frederick, MD 21702-5012

Commander
U.S. Army Medical Research and Development Command
ATTN: SGRD-PLC
Fort Detrick
Frederick, MD 21702-5012

Commander
U.S. Army Medical Research Institute of Chemical Defense
Aberdeen Proving Ground, MD 21010-5425

Commander
U.S. Army Chemical Research, Development and Engineering Center
Aberdeen Proving Ground, MD 21010-5423

Commandant
U.S. Army Chemical School
Fort McClellan, AL 36205-5020

Commander
U.S. Air Force School of Aerospace Medicine
Brooks Air Force Base, TX 78235-5000

Commanding Officer
Naval Health Research Center
P.O. Box 85122
San Diego, CA 92138-9174

Director
U.S. Army Laboratory Command
Human Engineering Laboratory
ATTN: SLCHE-SS-TS
Aberdeen Proving Ground, MD 21005-5001

Commander
U.S. Army Biomedical Research and Development Laboratory
Fort Detrick
Frederick, MD 21702-5010

Commander
U.S. Army Medical Materiel Development Activity
Fort Detrick
Frederick, MD 21702-5009

Defence and Civil Institute of Environmental Medicine
ATTN: U.S. Army Scientific Liaison Officer
(U.S. Army Medical R&D Command)
1133 Sheppard Avenue W.
P.O. Box 2000
Downsview, Ontario
CANADA M3M 3B9

1 Copy to:

Commandant
Academy of Health Sciences, U.S. Army
ATTN: AHS-COM
Fort Sam Houston, TX 78234-6100

Stimson Library
Academy of Health Sciences, U.S. Army
ATTN: Chief Librarian
Bldg. 2840, Room 106
Fort Sam Houston, TX 78234-6100

Director, Biological Sciences Division
Office of Naval Research - Code 141
800 N. Quincy Street
Arlington, VA 22217

Commanding Officer
Naval Medical Research and Development Command
NMC-NMR/ Bldg. 1
Bethesda, MD 20814-5044

Office of Undersecretary of Defense for Acquisition
ATTN: Director, Defense Research and Engineering
Deputy Undersecretary for Research & Advanced Technology
(Environmental and Life Sciences)
Pentagon, Rm. 3D129
Washington D.C. 20301-3100

Dean
School of Medicine
Uniformed Services University Of The Health Sciences
4301 Jones Bridge Road
Bethesda, MD 20814-4799

Commander
U.S. Army Aeromedical Research Laboratory
ATTN: SGRD-UAC
Fort Rucker, Alabama 36362-5292

Director
Walter Reed Army Institute of Research
ATTN: SGRD-UWZ-C (Director for Research Management)
Washington D.C. 20307-5100

Commander
U.S. Army Environmental Hygiene Agency
Aberdeen Proving Ground, MD 21010-5422

Commander
U.S. Army Military History Institute
Carlisle Barracks
ATTN: Chief, Historical Reference Branch
Carlisle, Pennsylvania 17013-5008

Commander
U.S. Army Natick Research, Development and Engineering Center
ATTN: STRNC-MIL
Technical Library Branch
Natick, MA 01760-5040

Commander
U.S. Army Natick Research, Development and Engineering Center
ATTN: STRNC-Z
Natick, MA 01760-5000

Commander
U.S. Army Natick Research, Development and Engineering Center
ATTN: STRNC-TAF
U.S. Air Force Liaison
Natick, MA 01760-5004

Commander
U.S. Army Natick Research, Development and Engineering Center
ATTN: STRNC-TAM
U.S. Marine Corps Liaison
Natick, MA 01760-5003

Commander
U.S. Army Natick Research, Development and Engineering Center
ATTN: STRNC-TAN
U.S. Navy Liaison
Natick, MA 01760-5003